JIT

Research Paper

Journal of Information Technology
1–26
© Association for Information
Technology Trust 2021
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0268396220978983
Journals.sagepub.com/jinf
SAGE

Distributed Internet voting architecture: A thin client approach to Internet voting

Jim E Helm

Abstract

Principles required for secure electronic voting using the Internet are known and published. Although the Internet voting functionalities and technologies are well-defined, none of the existing state-sponsored Internet voting approaches in use incorporate a total Internet-based system approach that includes voter registration, the voting process, and vote counting. The distributed Internet voting architecture concept discussed in this article uses a novel thin client approach to Internet voting. The architecture uses existing technologies and knowledge to create a viable whole system approach to Internet voting. This article describes various aspects and processes necessary to support an integrated approach. The application programming interface software for many of the critical functions was developed in Python and functionality tested. A virtual network, including a cloud-based functionality, was created and used to evaluate the various conceptual aspects of the proposed architecture. This included the concepts associated with programming and accessing smart cards, capturing and saving fingerprint data, structuring virtual private networks using tunneling and Internet Protocol Security, encrypting ballots using asymmetric encryption, using symmetric encryption for secret cookies, thin client interaction, and creating hash functions to be used within a blockchain structure in a Merkle tree architecture. The systems' primary user targets are individuals remotely located from their home voting precincts and senior citizens who have limited mobility and mostly reside in assisted living facilities. The research supports the contention that a cybersecure Internet voting system that significantly reduces the opportunity for mail-in voter fraud, helps to ensure privacy for the voter, including nonrepudiation, nonattribution, receipt freeness, and vote acknowledgment can be created using existing technology.

Keywords

Internet voting, electronic voting, thin client, e-voting, Merkle tree, blockchain, smart card, biometric identification, secure voting

Introduction

"For decades, the cybersecurity community has had a consistent message: Mixing the Internet and voting is a horrendous idea" (Parks, 2019). However, there is no unilateral consensus among the experts that Internet (or electronic) voting is impossible to protect and should be abandoned as a concept (Gibson et al., 2016). I contend that with today's technology, voting via Internet is no less secure or susceptible to fraud than voting by mail or maybe even in person. This is not to say that the vulnerabilities that exist in Internet voting or either of the more traditional voting approaches listed are similar. They are not. Each approach has its own type of vulnerability. No voting technique is perfect. For example, individuals' voting in multiple election locations across state lines is still feasible with both in-person and mail-in voting. Another example would be an individual voting in place of a registered voter who is incapacitated or deceased. There have been 1285 proven cases of voter fraud in America during the past 4 years (Samalis-Aldrich and VonSpakovsky, 2020). I believe this figure is ignored by the public because the perception is that this few numbers of cases cannot unduly influence the outcome of an

Arizona State University, USA

Corresponding author:
Jim E Helm, Arizona State University, 6049 S. Backus Mall, Sutton Hall 301M, Mesa, AZ 85212, USA.
Email: jim.helm@asu.edu

election. My point is that the majority of those who oppose Internet voting believe that the system must be perfect. History has shown that every voting technique has imperfections. The key to successful Internet voting is to mitigate the vulnerabilities and severely limit the probability of a successful cyber-attack. This article will identify and demonstrate how these two goals can be achieved.

Background

The vulnerabilities of Internet voting are well-documented. For example, most fall into one or more of the following categories (Jefferson et al., 2004a; Wolchok et al., 2012):

- Voter-verified audit trail;
- Insider attacks;
- Privacy;
- Vote buying/selling;
- Computer vulnerability;
- Voting environment control;
- Software application vulnerabilities;
- Man-in-the-middle attacks;
- Denial-of-service attacks;
- Server vulnerabilities;
- Network vulnerabilities;
- Voter coercion;
- Vote tampering;
- Voter Identification.

Privacy, vote buying/selling, voter coercion, vote tampering, and voter identification also exist in mail-in balloting. Two of the better known systems that are infamous for their security failures are the secure electronic registration and voting experiment (SERVE) (Jefferson et al., 2004a) and the Washington DC Internet Voting System (Wolchok et al., 2012). Wolchok et al. (2012) were a University of Michigan Computer Science team that conducted penetration testing on the Washington DC Internet Voting System. The system had numerous security problems on both the server side and the user side. The lack of security for the voter selections was one error. In addition, the server and its network had factory issued passwords for root access. With these vulnerabilities, the University of Michigan team could literally control all the balloting processes, including *stuffing the ballot box*.

One of SERVE's biggest vulnerabilities was its lack of ability to control the voting environment (Jefferson et al., 2004a). This led to many issues: polling setup, recording voter activities, voter monitoring, voting denial, unauthorized ballots, and privacy concerns. In other words, voter disenfranchisement was a viable threat. Other vulnerabilities within SERVE included spoofing and man-in-the-middle attacks. Finally, because of the architecture, undetectable denial-of-service attacks were also a vulnerability.

The purpose behind selecting these two examples was to show that the concerns and issues identified by Parks (2019) have not dissipated over time. There are still experts who rail against the use of electronic systems for voting (Scott, 2019; Vicens, 2019). These authors are using the same vulnerability issues highlighted in earlier articles related to Internet voting. The point is that there have not really been any new types of Internet vulnerabilities identified in the past decade; however, in general, the cyber community does not believe that the issues associated with Internet voting can or have been resolved. It is my contention that an Internet voting solution is a system problem that cannot be resolved piecemeal. In my review of the current literature, there are many very good ideas for protecting certain aspects of the voting process; however, most focus on a single aspect rather than an entire integrated system. I will reference these ideas in the appropriate section(s) of this article.

Current state of the art.

The use of asymmetric encryption in electronic voting is not a new concept (Al-Anie et al., 2011; Howlader et al., 2011; Meng and Wang, 2010). Over a decade ago, Alvarez and Hall (2004) discussed the future of Internet voting and the use of Internet voting in an Arizona county primary. However, there have been several detractors and failures of Internet voting systems due to technology and human failures resulting in several critics of using Internet voting technology (Davide et al., 2010; Dill and Castro, 2008; Epstein, 2013; Simons and Jones, 2012). One of the criticisms leveled at Internet voting systems is their lack of validation prior to deployment (Simons and Jones, 2012). For example, a Washington DC project was *hacked* by University of Michigan researchers in less than 36 h by exploiting several system vulnerabilities (Wolchok et al., 2012). One such vulnerability was that the master password had not been changed, resulting in root access by the penetration team.

Another commonly identified vulnerability is the computer used to interface between the voter and the Internet. However, advances in smart card technology and the use of Java capable smart cards as the primary interface have mitigated many of these type of concerns (Mohammadpourfard et al., 2015). The programmed smart card becomes the primary interface to the network instead of the computer, bypassing many operating system vulnerabilities (Mohammadpourfard et al., 2015).

The principles required for secure electronic voting using the Internet are known and published (Alvarez and Hall, 2008; Gerlach, 2009; Kaliyamurthie et al., 2013; Kavakli and Gritzalis, 2007). The security requirements for Internet voting are also documented (Beroggi, 2008; Mohammadpourfard et al., 2015; Neumann et al., 2016). At least six countries, Australia, Estonia, Canada, India, Norway, and Switzerland, are using or have used Internet voting technology to varying degrees (Anooja, 2016; Beroggi, 2008; Germann et al., 2016; Vassil et al., 2016). However, the exact details of the processes used in most of these systems have not been publicly disclosed by the respective governments. India is the most recent country to evaluate the use of Internet voting and is currently testing alternatives (Anooja, 2016). India has focused on using online voting for nonresident Indians and disabled voters. Only Estonia offers Internet voting to its entire population (Anooja, 2016).

Australia has been involved in some type of electronic voting for the past four decades (Buckland et al., 2012). However, similar to other Internet-based systems, the Australian system has had identifiable flaws (Buckland et al., 2012; Halderman and Teague, 2015; Stilgherrian, 2019). For example, client code was essentially accessible to the general public and there were issues with voting server configurations which could have led to reduced-strength cryptography (Halderman and Teague, 2015). In general, the most significant vulnerability was related to using local browser-based applications and its associated asymmetric encryption process (Teague, 2019). These system weaknesses imply that client-based voter applications (e.g. iVote and Swiss Post) where a voter locally selects his or her choices on a personal computer can have significant vulnerabilities.

Although the Internet voting functionalities and technologies are well-defined, none of the existing state-sponsored Internet voting approaches in use incorporate a total Internet-based system approach that includes voter registration, the voting process, and vote counting (Anooja, 2016; Germann et al., 2016; Vinkel, 2011). However, Adida (2008) did discuss the total system concept in his conference paper on a web-based auditing process that used homomorphic encryption. The proof-of-concept in this proposal will use existing technologies and knowledge to create a viable whole system approach to Internet voting. The uniqueness is in the innovative use of smart card technology coupled with a layered thin client architecture that uses both symmetric and asymmetric encryption and multilevel authentication with biometric identification. In addition, the current Internet voting schemes are relatively limited in scope (Germann et al., 2016; Vassil et al., 2016). The proposed architecture will evaluate economies of scale based on using network function virtualization (NFV) and software defined networks (SDNs). The use of biometrics for voter authentication has also been researched (Awad, 2011); however, the presentation was more a survey of techniques rather than an architectural approach. In addition, the relationship between end-to-end electronic voting and trustworthy computing has also been studied (Fink, 2010).

A robust, secure, and resilient architecture includes a common infrastructure with multilayer security, multilevel access, modular databases, control, authentication, and access using NFV concepts. This layered, distributed approach allows extensive reuse of architectural components for various disciplines while maintaining an ability to tailor data input and output to the needs of a particular architecture. The design includes extensible, scalable, distributed databases, each protected by a DMZ proxy server network with virtual servers. The design also incorporates smart card technology with biometric identification functionality. The proposed architecture consists of multiple virtual machines running different software and processes and incorporating a cloud infrastructure. This approach is typically identified as an NFV architecture. Cloud and multicloud databases are an extension of the fundamental techniques to protect application information. A good architecture design using SDN technologies can help defend against distributed denial of service (DDoS) attacks in a cloud-based environment (Wang et al., 2015). Big data technologies can be incorporated for large database structures. The multilayer security includes key infrastructure services that incorporate symmetric and asymmetric encryption, as well as cryptographic hash functions. The smart card data access requires biometric authentication. Layered encapsulating protocols are used to ensure data confidentiality during information exchanges. For example, the HTTPS protocol structure will be a foundational security feature that will be enhanced with other encryption techniques and protocols based on access type. Multilevel access includes authentication that incorporates biometric identification and microprocessor smart card functionality. In the context of this research, multilevel means that while the individual voter has access to the smart card in a read only context, he or she cannot access or make changes to the smart card. Another example of the multilevel access is that while a voter can access and vote a ballot in the cloud, he or she cannot access or modify any ballot features. Only certain administrators can access server functionality. The same is true for databases and other functions within the architecture.

Statement of the problem

Before Internet voting can be incorporated into the voting process, local, state, and federal bureaucracies must trust the viability of the system. The recent COVID-19 pandemic has contributed to the realization that senior citizens can be the most vulnerable to contracting fatal diseases in crowded environments. The strongest voting bloc (percentage wise) in recent elections was the 65- to 74-year-old age group followed by the 75 and up year-old age group (United States Census Bureau, 2017). We will not know whether the isolation as a result of COVID-19 disenfranchises this strongest group of voters until after the statistics of the 2020 election are known. However, it seems prudent to start evaluating whether an alternative system for senior citizens could be viable for those in care facilities as the population adjusts to this *new normal* existence.

The second group of voters who would benefit from an electronic-based Internet voting system would be military and other government employees who, due to assignment,

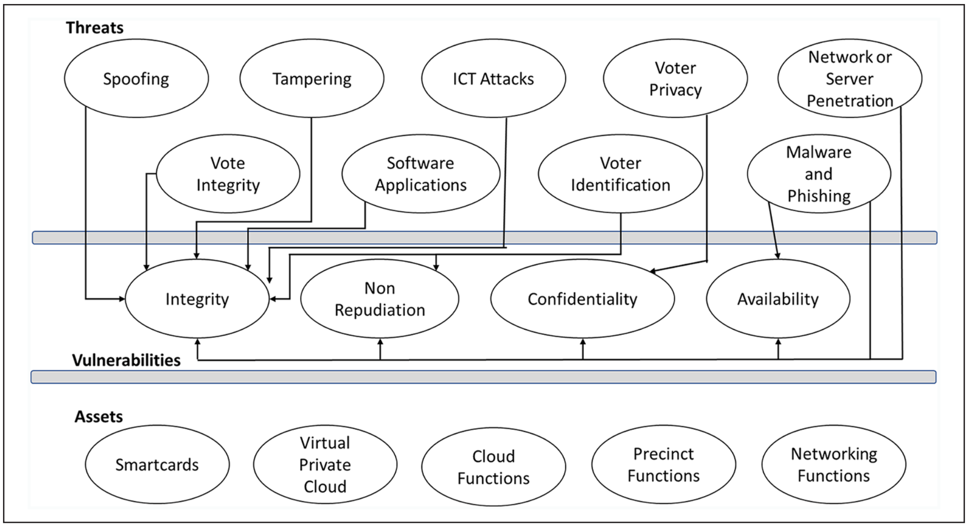


Figure 1. DIVA threat model.

are remotely located from their local voting district. There were about 157.6 million registered voters for the 2016 election. Approximately 8% of registered voters did not vote for various reasons. Approximately 12.6% of these registered nonvoters were not able to vote because they were away from their local voting district (File, 2018); 12.6% of registered nonvoters is equivalent to approximately 1.5 million individuals. Considering there are approximately 1.3 million service members and 68% were registered for the 2016 election with only 43% of those members voting, these statistics indicate that there is disenfranchisement among the military voters (FVAP, 2017). While these numbers may not seem to be significant, keep in mind that in the 2016 presidential election, the outcome of the election hinged on 107,000 votes in key states out of the total of 120 million votes cast (Meko et al., 2016).

Acceptance of a new technological approach often varies with the characteristics of the technology, the perceptions associated with the technology, and the benefit(s) associated with its intended use (Carter and Campbell, 2011; Carter et al., 2011). There are three factors that must be satisfied prior to the time when Internet voting will be accepted by all the stakeholders: societal acceptance and trust, political considerations, and cyber community technical acceptance (Carter, 2006; Chevallier, 2009). In addition, studies have shown there are factors that influence technology acceptance by a general population: Technology Acceptance Model (Davis, 1989); Diffusion of Innovations (Rogers, 2003); Unified Theory of Acceptance and Use of Technology (Venkatesh et al., 2003); and The Utilization of e-Government services: Citizen trust, innovation, and acceptance factors (Carter and Bélanger, 2005). There are myriad studies that delineate the models and processes associated with acceptance of using new approaches (including Internet voting), based on these foundational studies. I believe acceptance of distributed Internet voting architecture (DIVA) by the technical community is most influenced by trust because of historical flaws in Internet voting systems. Trust of the Internet and its cybersecurity capability to protect Internet voting data from attack is a valid concern. This means extensive penetration testing of the DIVA architecture would be required to alleviate vulnerability uncertainties. An in-depth analyses of the factors influencing acceptance is beyond the scope of this article. However, my contention is that the cyber community technical acceptance is the most important impediment to overall acceptance because research has shown that societal acceptance by senior citizens will occur if they are provided the Internet voting technology (Helm, 2015).

Therefore, if the cyber technical community can be convinced that there is a secure approach to Internet voting, then this electronic voting system can become a reality for limited populations (as a beginning). This article introduces an Internet voting concept that addresses major failings of previous approaches, as well as, anticipating and mitigating threats to a networked system. Keep in mind that voting is a short-term activity. In other words, the goal is to design and build an impenetrable system over a short duration of time because after votes are cast and counted, system penetration becomes moot.

Threat assessment. The threat model for this architecture is shown in Figure 1. The model in Figure 1 is based on a visual, agile, and simple threat modeling (VAST) process (Walker, 2019). VAST was selected as the model of choice because the derived workflow diagrams that illustrate threats, assets, vulnerabilities, and remediation tools can model both the operational and application threats. Only the VAST top layer is shown. The three sections associated with this model are: threats, vulnerabilities, and assets. The threat profile is based on previously identified threats published in the literature (Al-Ameen and Talab, 2013; Eom et al., 2015; Scott, 2019; Smith, 2018). The vulnerabilities are the common vulnerabilities faced by any Internet-based system: confidentiality; integrity; and, availability plus nonrepudiation because of the purpose of the architecture. One vulnerability to integrity external to both the client machine and the smart card is the fingerprint reader. The possibility that the same fingerprint reader could be used during both registration and voting and that this device was spoofed to provide bad data is possible. However, it is unlikely. Spoofing a single fingerprint is not effective and if an attacker spoofed multiple fingerprints by replacing them with his or her own, this would be detected when the voter registration was validated and the fingerprint data verified. The registration would be rejected. The assets that necessitate protection associated with this system require a more in-depth discussion.

Smart cards—This contact-based secure microcontroller smart card includes nonvolatile memory, RAM, ROM, user memory and input/output functionality. The IC is programmable with dynamic functionality. The DIVA specific systems and data are stored in nonvolatile memory. Dynamic active security is used for authentication purposes. The card is programmed with the user fingerprint data for this purpose. Additional data include AES symmetric encryption key for FIDO interactions, precinct public key for asymmetric encryption, crypto processors, user registered email address, user voter registration number, user name, state IPv6 address for VPN cloud, precinct IPv6 address for ballot, authentication IPv6 address for two-factor authentication, social security last six digits for pin and SHA256 hash functionality. Embedded within the smart card are various security functions to mitigate tampering. The smart card complies with ISO/IEC 7816 parts 1, 2, and 3. The smart card incorporates hardware, software, and system countermeasures to protect data and interchanges because no single security measure can protect against a broad spectrum of attacks. The Smart Card Alliance describes these security protections in detail (Smart-Card-Alliance, 2008). The security features of this smart card make it highly unlikely that a smart card could be penetrated or altered without either being discovered or destroying data stored on the smart card. Depending on the manufacturer, smart cards have the ability to delete stored data if penetrated without proper authentication.

Virtual Private Cloud—The VPC for this network could be either leased or government owned. There are two types of Proxy Gateways in this architecture: Voter accessible and Administrator accessible. There are 50 voter accessible IPv6 Proxy Gateway accesses; one for each state. There are also 50 administrator accessible IPv6 Proxy Gateway limited access (based on IPv6 address); one for each state. Part of the distributive nature of this architecture is that each state has its own IPv6 Gateway into the cloud functionality. All of the DIVA Gateways require tunnel VPN ICT. FIDO authentication is required as part of the VPN tunnel interconnectivity and access to hidden VPC servers. This means that if an attacker wants to disrupt voting on a national scale, he or she would have multiple targets that would require disruption. Here again, the cost/benefit ratio becomes a factor.

Cloud Functions—These include SDN functionality, authentication functionality, thin client ballot functions, temporary data storage, voter processes, and voter notification. The protected servers within the VPC have limited access. For example, while the voter has access to his or her precinct ballot for casting a ballot, he or she does not have access to the ballot server that generates the ballot. Only the administrator, after authentication, has access to program the ballot server. In addition, the ballot server holds only the compiled version of the ballot creation software. The source code is not stored in the cloud. All of the internal processes, databases, and operational functions are protected by limited access. Certain sensitive information is encrypted. For example, when a voter completes casting his or her ballot and submits it via the thin client, the results are encrypted using the precinct public key. The ballot is not decrypted until it arrives at the precinct location and it is decrypted off-line. The ballot is also saved in a hashed blockchain for audit and anti-tampering protections. The VPC functions also include authentication functions. The system requires two-factor authentication prior to initiating the voting process and this capability includes tracking voter identification to preclude multiple votes from a single voter. Prior to casting a ballot in the thin client architecture, a voter is also authenticated using a Kerberos process. This allows direct interaction between the voter and the vote casting process. This process frees the ballot server to service multiple ballots simultaneously. Many of the web server application programming interfaces (APIs) associated with cloud process and database functions and controls are based on the Django framework. Bundled Django based distribution supports a framework that allows it to run multiple websites. This supports the SDN architecture within the cloud functionality. Django incorporates mitigations for cross-site scripting, SQL injection, plus many other security measures (Django, 2020).

Precinct Functions—In the concept of this article, a precinct is a loosely defined voting subdivision. The actual structure of a voting precinct can vary state-by-state. It could be based on county boundaries or metropolitan boundaries or state or federal congressional districts. This means that the number of precincts will most likely vary state-to-state. However, the functionality required by the precinct remains consistent across all states. The precinct houses the ballot application software for all its voters. The precinct creates the ballot offline and then saves it to a secure USB which is uploaded to the VPC via the VPN tunnel ICT. Structuring the system so that the ballot is only created in an isolated environment improves security. The precinct also conducts any audit functions and checks to ensure that the downloaded ballots do not indicate tampering. The precinct also separates the individual ballots and transfers them to a secure USB so that they can be decrypted offline and entered into a vote-counting machine. The private key for the precinct is housed offline and is used to decrypt the incoming ballots prior to counting. The private key is also used during ballot uploads to create a digital signature for verification purposes. The precinct also houses its voter information, including the voter registration data with fingerprint records in a segregated offline database. This protects sensitive voter data from attacks and precludes the likelihood that an attacker could access fingerprint data and alter voter files. After a new voter completes registration, his or her data are transferred to this offline database. To further protect these data, a two-person access rule should be incorporated. As part of the voting process, the administrator uploads hashed voter information which will be used to compare to hashed voter information created in the smart card and transferred during ballot casting. This is part of the voter validation process. The process protects the privacy and identity of the actual voter while supporting nonrepudiation and nonattribution functionality.

Networking Functions—This asset includes the functions required to successfully move data from a source to a destination. The structured ICT process incorporates encryption to protect confidentiality, authentication to preclude unauthorized users, and integrity to detect any tampering. At the network layer, Internet Protocol Security (IPSec) is used to hide the final destination, limited access IPv6 address. The client-server connectivity uses TLS 1.3 for VPN. Prior to ballot access in this thin client architecture, a Kerberos process is required. This process allows the voter direct access to the ballot within the VPC after authentication and ticket granting. This process eliminates the need for the voter to have access to the ballot server. Initial authentication is based on the FIDO Alliance in conjunction with the smart card which contains the fingerprint information. All accesses to the VPC by any voter or administrator travel using VPN tunnel functionality. Administrator interactions are also further protected by using IPv6 addressing filters that limit VPC access to a given set of IPv6 addresses for a given IPv6 VPC Gateway.

Risk analysis. In any risk assessment, it is important to understand the risk profile for the architecture being evaluated. For Internet voting, the dilemma for the attacker is to understand effort versus reward. In this proposed distributed thin client architecture, an attacker starts with a limited timetable. In a hypothetical case, let us assume that the window of opportunity for the attacker is 1 week. This means that he or she has to penetrate all the layers of security and encryption within a distributed architecture in about 1 week. For the sake of argument, suppose an attacker is able to penetrate a user's smart card. Two red flags would be created: (1) many smart cards with IC microcontrollers erase data when penetrated and (2) any modifications to critical data would create a situation where the hash functions would no longer match. Also, when one considers the fact that a successful change to a smart card would influence one vote, the reward versus the effort becomes questionable. This architecture also helps mitigate the possibility of extensive voter coercion. It is true that a single voter could be intimidated or coerced into voting a specific way. However, for a coercion campaign to be successful, there would need to be many victims in order to influence an outcome. This same type coercion vulnerability exists with mail-in voting. However, unlike mail-in voting, techniques such as ballot harvesting, unauthorized voter ballot submittal, and mail-in ballot acceptance and counting irregularities are not possible with this architecture. The architecture also blocks any attempt to cast multiple ballots or ballots from deceased persons. Once an individual casts a ballot that is accepted by the system, he or she is blocked from submitting another electronic vote and his or her data are recorded so that an attempt to vote in-person triggers a conflict.

The second area of potential attack would be the VPC. Several characteristics of the DIVA architecture also make this approach difficult for the attacker. First of all, cloud access points are distributed and vary by state. Second the VPN tunnel structure makes it difficult for an attacker to gain access. The IP addresses Access to the ballot requires a Kerberos-based authentication process. This means that the attacker would first have to have defeated the smart card protections in order to successfully complete the authentication process. The thin client architecture also means that the voter access to the ballot acts like a remote display where the voter makes his or her selections. Gaining access to the ballot would not create multiple incursions. The ballot is isolated from the ballot server and the ballot cannot transfer unencrypted data to the server. Here again, the time frame and the distributed nature of the architecture makes it difficult for the attacker.

The third area of potential attack would be the precinct that houses the software and counts the votes. The first area of consideration is the actual ballot. The application software that generates the ballots is not connected to any external network. Similar to the way ballots are generated in current configurations, the ballot data are created and then transferred to a secure USB. While the application that interprets the ballot does reside in the cloud, it must be programmed with compiled software in order to generate the ballot. Each precinct will transfer these data via VPN tunnel connectivity to an IPv6 address in the cloud from its administration IPv6. This means that it would be very difficult to access ballot development software. To protect against insider attacks, accessing the isolated ballot software requires two-person control. The limited access interface between the VPC and the precinct with two-factor authentication makes it very difficult for an attacker to penetrate either the VPC or the precinct through these communication portals. Each uploaded ballot incorporates a digital signature to ensure that the ballot has not been altered during transit. After being downloaded, the incoming data are checked to ensure the blockchained ballots do not show any aspects of tampering. These ballots are then saved to a secure USB and moved to a closed system for decryption and counting. The downlinked data are also saved to databases for audit purposes. Because most all of the ballot development and counting application software are operated in a closed environment, the potential for outside penetration is nearly impossible.

In summary, the limited time frame a voter has access to the system and the distributed nature and complexity of the system makes attacker penetration extremely difficult. The amount of resources required within the given time frame versus the likelihood of influencing the outcome through tampering with an election significantly mitigates an attacker's motivation.

Table 1 shows a threat summary for the DIVA architecture. Notice that it incorporates most all the standard threats to Internet voting identified earlier.

System integrity. System integrity is a key characteristic of any electronic voting system (Awad, 2011; Mohammadpourfard et al., 2015). The decade old Secure Electronic Registration and Voting Experiment (SERVE) sponsored by the U.S. Department of Defense identified many vulnerabilities (Jefferson et al., 2004b). Previous research has assumed that an adversary would not be able to simulate a voter during registration (Clarkson et al., 2008). Incorporating biometrics significantly reduces the likelihood of fraudulent registration using a different person's name. Vote integrity is another area requiring special attention. Cryptographic hash functions can be used to validate voted ballots. However, in order to maintain integrity and preclude ballot modifications, Merkle trees will be used. Merkle trees ensure that individual ballots have not been modified by ensuring that the hash sequences in a list have not been altered (Bakker et al., 2015). Each state has its own Merkle tree with the branches being the voting district (e.g. Congressional district) and the leaves being the various voting precincts within the state. The use of Merkle trees in this cloud-based architecture also precludes packet dropping attacks, providing confidence that data are forwarded without loss (Ahmad et al., 2016; Mao et al., 2015). Therefore, ballots will include cryptographic hash functions for audit purposes. The leaves are constructed using a mineblock() Blockchain with nonce that contains the encrypted ballots. This approach should work well for this large distributed system (Mao et al., 2015). One of the inputs to the hash function will incorporate the current, prevoted trusted state of the ballot. These measures help ensure the integrity of the ballot and provide voters with confidence that their vote has been counted. An email receipt is sent to each voter when he or she submits his or her ballot indicating whether the voting process was successful or unsuccessful. This receipt allows the voter to confirm whether his or her ballot was counted. If the vote was rejected, the system allows the voter to try again because the process does not record a successful voter submittal.

Multiple vendors now support SDN centralized management and control and provide common application programming interfaces (APIs) that increase network reliability and improve security (Bailey et al., 2012). The cloud offers new challenges and moving IT infrastructure to the cloud also poses new threats. The cloud can offer scalable processing and an SDN architecture adds network management flexibility at a reduced cost (Wang et al., 2015). For example, using an SDN architecture can help mitigate DDoS attacks (Wang et al., 2015). SDN offers the ability to enforce and manage multiple security policies in real time because of software control (Haleplidis et al., 2015). Similar to all other systems, the management control for the system architecture within the VPC requires two-factor authentication prior to access.

Research question. The primary research question for this effort is: "How can the concept of a cybersecure Internet voting process that incorporates voter integrity, nonrepudiation, voter privacy, and accurate vote recording with nonattribution be architected?"

Hypotheses

H1. A cyber-attack resistant cyberinfrastructure can be instantiated in a multilevel, multilayer, distributed virtual system using modified web server technologies for VPC server access and processes within the modern Internet Communication Technologies (ICT) architecture.

H2. The cyberinfrastructure incorporating smart cards and biometric identification offers a cybersecure network that is resistant to unauthorized access, man-in-the-middle attacks, malware, and denial of service attacks.

H3. An Internet based cybersecure registration and voting system cyberinfrastructure can be implemented in a modular architecture using existing technology.

Nature of the research

The goal of this research was to create a cyber secure concept that could be used to allow safe and secure Internet voting for certain populations within the voting populace. Starting with a study of previous systems and inputs from cyber security experienced individuals; vulnerabilities associated with Internet voting were evaluated. Using this information, a conceptual objective to solve the vulnerability problems was developed.

Table 1. Threat summary.

Threat	Property	Mitigation
Computer malware	Integrity	• Smart card-based OS • Thin Client architecture
Spoofing and tampering	Integrity	• Smart card w/secure microcontroller IC • Smart card-based biometric authentication • Digital Signatures • Asymmetric Encryption • Two-factor authentication incorporating FIDO functionality • Functional Diversity—Multilevel Access • SHA-256 Hash Functionality
Verifiable Audit Trail	Nonrepudiation	• SHA-256 Hash Functionality • Segregated Databases • Digital Signatures • Vote confirmation
Man-in-the-Middle Attacks	Integrity/Availability	• VPN Tunnel communications • Symmetric Encryption
Denial-of-Service Attacks	Availability	• VPN Tunnel communications • Distributed VPC access
Voter Privacy	Confidentiality	• Thin Client Architecture • Asymmetric Encryption of ballot • VPN Tunnel communications • Segregation of voter identification and ballot • Hash function of header that includes voter information
Network and Server Vulnerabilities	Availability/Integrity	• VPN Tunnel communications • Distributed servers • Distributed VPC access IP addresses • Limited administrative access via separate authentication structure • Network function virtualization • SDN architecture • Distributed Proxy servers • Distributed NF V
Software Application Vulnerabilities	Integrity	• Smart card w/secure microcontroller IC for voter-related operations • Use of compiled software programs in most cloud-based functions • Limited access to software application code • VPN only access to cloud-based servers between limited access VPC Gateway and Precinct ballot compiled software
Insider Attacks	Integrity	• Two person only access to sensitive systems and programs • Audit process based on blockchain hash functions and asymmetric encryption
Vote Integrity	Integrity	• Asymmetric encryption of ballot where private key is owned and controlled by voter's precinct and is never stored or used outside of a secure facility • Blockchain hash sequences with nonce for integrity of chain • Thin client only access to cloud-based ballot using Kerberos authentication • Limited time frame vulnerability
Nonattribution	Confidentiality	• Use SHA256 hash function to capture voter header information • After ballot is cast segregate all voter information from ballot and encrypt ballot • Separate databases for voter information and ballots

Practical application of a theory means the research contributes to operational performance knowledge (Venkatesh et al., 2013). One of the purposes of research is to contribute to filling a void or to extend the understanding of existing concepts. This research is intended to show a concept that mitigates the threats and vulnerabilities to Internet voting.

Direct Recording Electronic (DRE) Systems have been in use in the United States for most of the twenty-first century. Many of these devices have become antiquated and are in need of replacement (Cortes and Norden, 2018). Many manufacturers have products that could be integrated into the system architecture. Although, these are normally stand-alone devices at this time, they are typically uploaded with software before an election via a secure USB. This process could be completed via a VPN connection into a cloud based thin client architecture. The USB relevant data would be uploaded to the VPC via IPv6 tunnel by the precinct administrator. Many manufacturers' systems use a Windows operating system foundation for their applications. For example, Election Systems & Software and Unisyn Voting Solutions are two companies who meet these criteria. The United States Election Assistance Commission (USEAC) maintains a complete list of authorized voting equipment manufacturers (USEAC, 2020).

The approach taken by this research was to capture an integrated system view of the threats and vulnerabilities associated with Internet voting and develop mitigation practices that address all expected variations of this threat. Secure communications is one aspect. For example, all external communication exchanges use virtual private network (VPN) tunneling techniques, in addition to other symmetric, non-symmetric, and hash encryption processes. The VPN interacts with a virtual private cloud (VPC). One of the key attributes to this concept is the distributed nature of the sources and destinations. The secure VPN Internet Protocol (IP) address identifies a secure gateway. The secure gateway is the only server that has access to internal VPC servers. Each state has its own secure gateway IPv6 address with limited VPN access.

Another vulnerability associated with previous systems included the lack of authentication processes. Single-factor authentication can be vulnerable to many attacks, including, but not limited to, password theft, password dictionary attacks, forgery, etc. This concept uses two-factor authentication. Two-factor authentication is more secure, especially when it is coupled with biometric identification. This concept uses biometric identification as one element in the security toolbox arsenal. Other technologies that will be addressed in this concept include a Merkle tree process within a Blockchain structure that links voter responses using hash functions.

There are two fundamental threats to any Internet-based system: computer vulnerabilities and network vulnerabilities. Computer vulnerabilities include viruses, worms, Trojan horses, phishing targeting, logic bombs, etc. Network vulnerabilities include MAC flooding, ARP spoofing, network taps, port and vulnerability scans, cross scripting, code injection, denial-of-service attacks, network intrusion attacks, etc. To address these and other vulnerabilities, the following functions have been integrated into a single concept:

- Smart cards;
- Biometric identification;
- Asymmetric encryption based on X.509 standards;
- Symmetric encryption;
- Hash functions;
- Two-factor authentication;
- Virtual private networks with tunneling using IPSec;
- Merkle tree process in a blockchain structure;
- Virtual private cloud;
- Distributed databases;
- Distributed gateway servers;
- Segregated applications.

The target audience for this research is those who would be interested in a proof-of-concept program that includes all of the infrastructure in addition to a complete penetration testing procedure. The initial pilot program after testing would target military and other personnel deployed outside of the country. This system will enable stakeholders to remotely cast their votes via a secure architecture with minimal risk of compromise.

Research objectives. These objectives support the overall expectation that this multilevel, multilayer, distributed cyber secure infrastructure will demonstrate that a viable Internet voting system can be developed and validated. The uniqueness of this comprehensive approach is in the use of smart cards with biometric authentication and the multidimensional structure encapsulating multipurpose objectives in a thin client architecture. This means interactions between clients and servers vary based on the role of the client, eliminating unnecessary exchanges. This approach should help to create a cybersecure architecture that is more efficient. Demonstrating the proficiencies associated with this proposed approach should significantly add to the body of knowledge related to using the Internet for voting within a cybersecure cyberinfrastructure by introducing a new and unique approach to meeting voting objectives: proven security, vote verification, voter accessibility, voter anonymity, very low probability of voting fraud, and vote recording accuracy.

> *Objective 1.* Create a multilevel authentication process that incorporates smart cards and biometric identification in order to dictate who and how authorized access is implemented.
>
> *Objective 2.* Use multilayer encryption at the Transport, IP, and Application layers to facilitate interactions among clients, control servers, database servers, cloud servers, and protected databases, including a structure that incorporates a receipt protocol for voter verification and an audit function for vote integrity.
>
> *Objective 3.* Conceptually demonstrate how middleware control software and application programming interfaces (APIs) can be created to isolate and protect sensitive information contained in a database.
>
> *Objective 4.* Conceptually demonstrate how protected database information can be shared with a platform as a service (PaaS) cloud database for general public voting purposes in a thin client architecture without compromising sensitive data.

Significance of the study

There have been several research studies related to electronically driven functionality in government applications (Bélanger and Carter, 2012). Internet voting is a topical area that has been investigated by evaluating different functions within the architecture. Much of the recent literature still highlights vulnerabilities identified in the early 2000s (Chipman, 2016; Sabin, 2018; Scott, 2019; Vicens, 2019). Rosacker and Rosacker (2012) found that advancements in information communications technologies (ICTs) provided a solid foundation for remote access voting; however, at the time, there had not been enough progress to overcome issues and concerns. I submit that at this time, there has been enough progress to warrant a proof-of-concept demonstration of the proposed approach that includes penetration testing.

This research identifies a concept that incorporates an integration of myriad technologies that currently exist which can be used to mitigate the vast majority of threats associated with Internet voting. The concept is a DIVA. Similar to the Internet, the network architecture is complex while maintaining transparency for the casual user. Individuals need to know the fundamentals of *driving* the computer, but they have no need to understand how it works within the network architecture in order to use the system.

This concept is the first to integrate the Internet voting method described in a patent created by Chung et al. (2005). Chung, et al. proposed a packet-based voting scheme using biometric identification. While there are several security issues with the authors' approach, the basic theory is sound. This integrated concept builds on that approach while incorporating and integrating into the architecture security features that enhance the feasibility of this patent. The timing of this research is important because there have been several recent news reports about concerns related to mail-in ballots. This approach eliminates many of the concerns regarding voter fraud associated with senior citizens and mail-in voting. Also, the last extensive testing that documented Internet voting vulnerabilities was over 8 years ago by Wolchok et al. (2012). Cybersecurity has made significant advances over the past 8 years (Lawyer Monthly, 2019). This research is the first step in showing how these advances can lead to a viable Internet voting system.

System architecture

The DIVA project presented in this article is based on individual research at Arizona State University (ASU). The DIVA sets out to create a distributed, secure, and reliable voting architecture for targeted eligible voters in the US elections. Several entities have tried to implement online voting architectures that fall short in areas, such as security, implementation, and ease of use. The approach outlined in DIVA is different as it incorporates multiple layers of security, including biometrics, encryption, two-factor authentication, and utilizing smart cards to authenticate the voter and allow access to the application.

The architecture incorporates many functions: infrastructure management; databases; server deployment; encryption, hashing, and encoding; two-factor authentication using biometrics; VPN tunneling; cloud computing; web-based interactions; and smart cards.

Overview

This concept provides a secure voting system that uses Internet security approaches similar to online banking or crypto-currencies. The concept uses a thin-client structure over an encrypted, secure link where ballots are completed online and submitted with a digital signature to ensure authenticity and non-repudiation. In addition, the areas of confidentiality, integrity, and availability must be included. Some of the keys to the success of this program are voter registration, data integrity, voter confidentiality, and ease of use.

Theoretical framework

Figure 2 shows the conceptual block diagram for the architecture. For example, an access request to a particular server using its uniform resource locator (URL) or IPv6 address would first encounter the appropriate security layer. All general public interactions are forwarded to the interface and control where an exchange between the devices completes a health check and instantiates an encrypted link between the layered information communications technology (ICT) infrastructure and the control. The interface and control, acting in the vein of a proxy server establishing a DMZ and using an NFV infrastructure, establishes an encrypted and authenticated link between itself and the appropriate cloud database. The cloud database interacts with the voting system databases. Encryption functionality and authentication is used in all interchanges and on the smart card. The initial interface and authentication control include a trusted infrastructure based on FIDO concepts. The client computer uses a digital certificate located in the smart card to begin the authentication process. The client request is also used to establish a VPN between the client and the cloud access database. The security and authentication are layered. For example, during the voting process, smart card validation using biometric identification is required prior to beginning the voting process, while during registration, biometric data are collected and forwarded to the registration database for confirmation and recording. Smart cards are programmed by the system owner and provided to registered voters. At the time of registration, the voter information will be stored in a provisional database. This would mean that a voter could vote a provisional ballot until he or she receives his or her smart card. Provisional balloting could also be used in the event a voter had his or

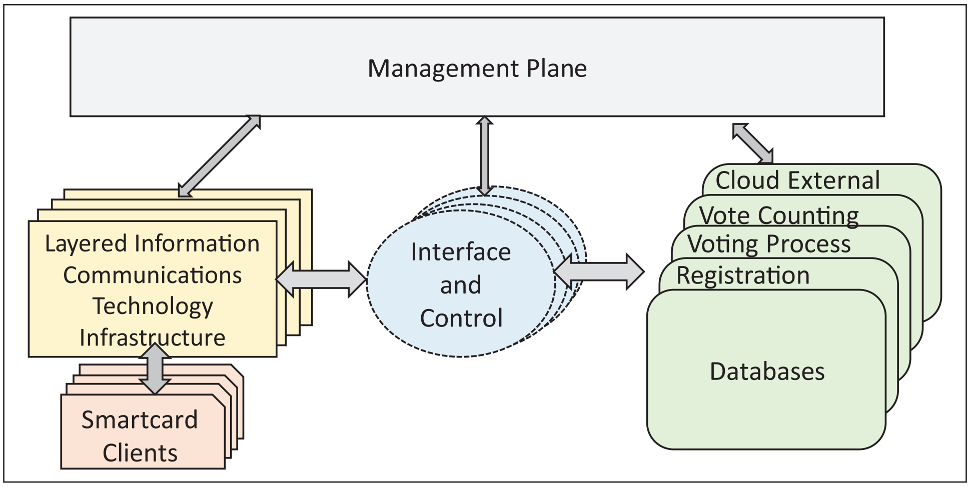


Figure 2. DIVA architecture based on software defined networking concepts.

her smart card stolen, damaged, or lost. The voter data are confirmed by a system administrator prior to moving information from the provisional database to an encrypted registered voter database. Most of the system administrator functions can be automated.

Each of the primary planes in Figure 2 is layered and its particular functionality is based on its role in the transferring of data from a source to a destination. Connectivity between the interface and control and the related databases requires enterprise application integration (EAI) incorporating newly developed layered object request broker middleware. The middleware will act as a firewall between the requestor and the database; setting the security parameters and limiting access to specific databases and database tables. Depending on the process, this could include database replication into a cloud or multi-cloud environment to protect the sensitive data from tampering. This means that voters do not have direct access to protected databases. All voting process actions interface with a cloud-based system.

Proposed architecture. Figure 3 shows the secure Internet voting system architecture. There are three subsystems associated with any voting system: voter registration, voting process, and vote counting. This system integrates all aspects associated with voting into a single architecture. Voter registration includes collecting the biometric data and demonstrating proof-of-citizenship. The voting process incorporates functionality required for a secure system, including voter identification, ballot dispersal and collection, vote confirmation, ballot and voter recording (separately), and voter nonrepudiation. Vote counting is the process of counting the ballot selections and recording the outcome. Figure 3 shows the three characteristics associated with the voting process. The system architecture includes all aspects associated with voting within an integrated architecture.

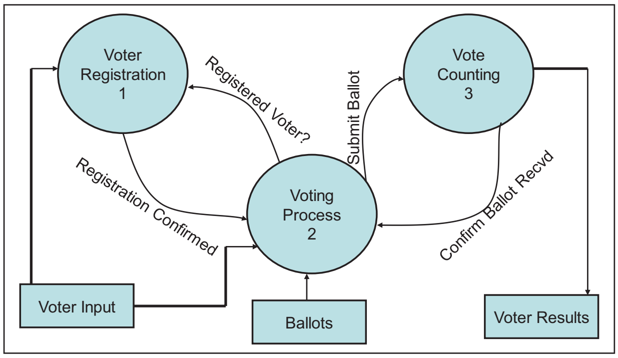


Figure 3. Internet voting architecture.

Voter registration. The goal of this process is to ensure the activity is simple and relatively easy for the voter while maintaining the integrity of the voting process. An individual could complete the registration online by himself or herself or with the help of another as long as he or she has the registration application on his or her computer. However, biometric (fingerprint) data are required as part of the registration process; which means a fingerprint reader is required in order to register to vote. The binary fingerprint data file is converted to Base64 for storage and transmission. The fingerprint data are transferred as part of the registration data. These data are hashed using a SHA256 process for validation purposes to mitigate the possibility of tampering during transit. The registration collects the relevant information about the individual, including some type of proof-of-citizenship documentation (e.g. passport #, birth certificate #, naturalization #). The registrant also would indicate whether the device being used for registration would be the same device used in the voting process. If the registrant answers yes to this question, then the device is checked for compatibility and if compatible, the device information is saved with the registration data. All of the data are encrypted before transfer and submission using TLS 1.3.

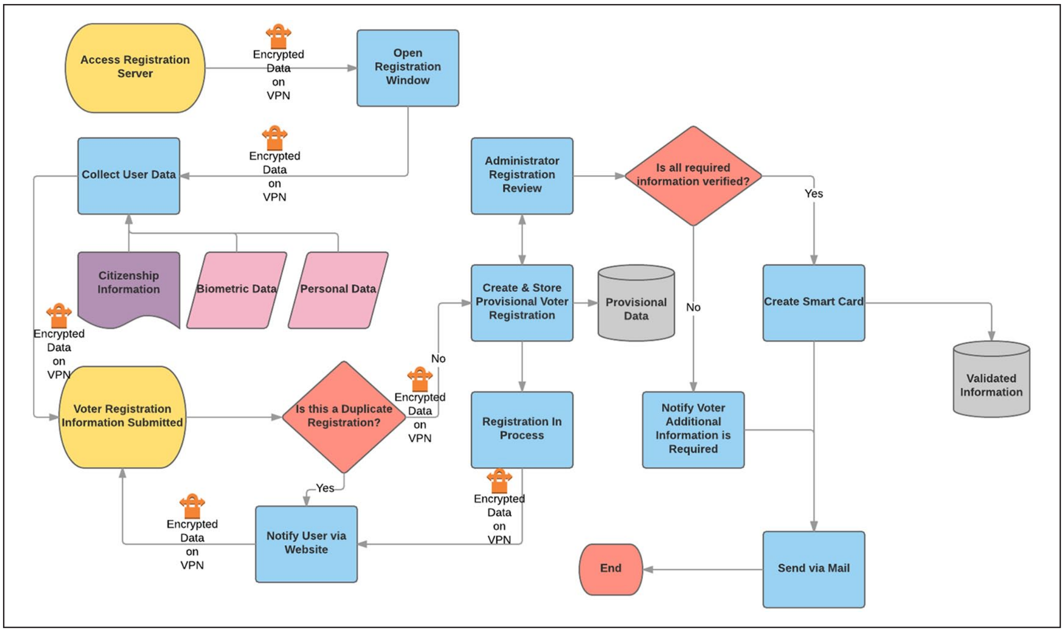


Figure 4. Voter registration process.

The registration information is forwarded to an administrator who verifies the data (e.g. citizenship) and checks for duplicate registrations. The administrator then transfers the data into an offline, secure, partially encrypted database. MariaDB structured relational databases are used for most database systems in this architecture. At the time of registration, a voter identification number is assigned. This value becomes the primary key for the databases that contain voter data. The social security account number (SSAN) is a secondary key for each registered voter. A microprocessor capable smart card with important identification data, encryption functionality, and biometric information is programmed and mailed to the individual and the registration process is complete. When the voter registers and prior to information confirmation, his or her data are stored in a provisional voter database. This means that as soon as a voter has completed the registration process, he or she could submit a provisional ballot. Provisional ballots could also be used by voters who have lost (or had stolen) their smart card or damaged the smart card; although the identification data would have to be retrieved from the registered voter database. Figure 4 shows the process voters will encounter during the registration process. Notice that an initial check for duplicate registration is completed at the time of registration. This is based on provided information. Also, registering voter data are captured in a provisional database to allow voting to occur during the vetting process.

Voting process. This process is the heart of the Internet voting proof-of-concept architecture. The goal is to have a cybersecure cyberinfrastructure that is relatively easy for the average voter to comprehend and use. The proof-of-concept system will incorporate a Windows 10 operating system (OS) with the Fast Identity Online (FIDO) Alliance version 2.0 in conjunction with a smart card. The World Wide Web Consortium (W3C) has completed development on an open standard that uses the Fast Identity Online (FIDO) Alliance version 2.0. The FIDO Alliance has created a set of protocols that protect user privacy through biometric identification (FIDO Alliance, 2014; Shamas, 2018). The W3C has adopted most of the FIDO standards (Bharadwaj et al., 2016). The standard is supported by both browser providers and operating system providers in their latest versions and can be implemented when the standard is completed and released (Jin et al., 2015). Fundamentally, the biometric information resides on the smart card. When a user confirms his or her identity, a protected file is unlocked. This allows the client to interact with the server and provide credentials showing that the interaction was initiated by an authorized user. The authorized user is verified by the server using the encrypted cookie. Only the server can decrypt the cookie. The user public key is stored as a field in the cookie, allowing the server to access the client information. The server also compares the received data to a table showing that the user and client are registered and approved. Figure 5 shows the process voters will encounter during the voting procedure. Notice the smart card contains the information needed to complete the initial voter authentication. After the voter submits his or her ballot, it is asymmetrically encrypted with the precinct public code stored in a blockchain Merkle tree structure, and a confirmation email is sent to the voter that includes

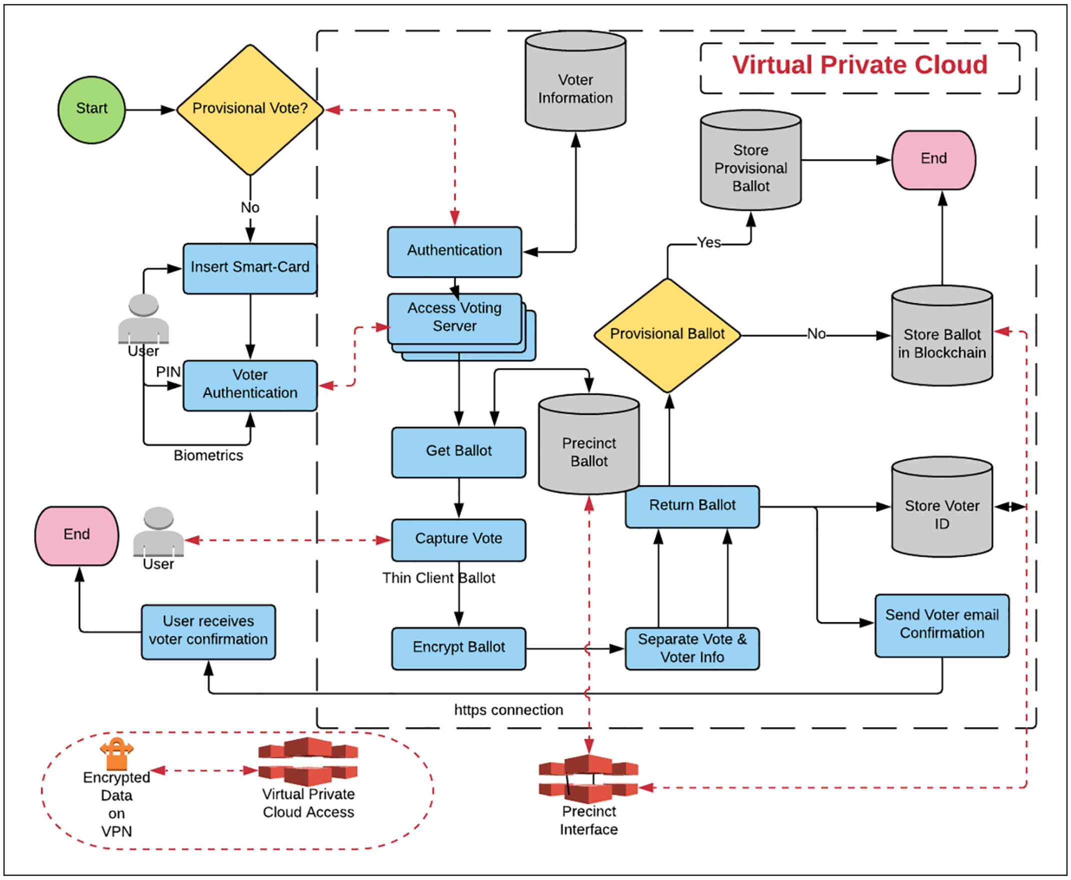


Figure 5. Voting process.

data identifying the election and a receipt that confirms the voter's ballot was accepted.

Registration is required prior to information exchange. The voting process also incorporates IPSec at the network layer to establish a protected tunnel for communication between the client and the control/DMZ server. The encrypted cookie returned by the client was placed on the client smart card during the registration process. The cookie includes unique identifiers, including cookie and client identifications, registration approval, device public key, device credentials, biometric information, and an expiration date. Registration ensures that a particular client is an authorized user and smart card device prior to any database interaction. Authenticated devices will not have access to all tables or objects. The interaction between the voter client and the Internet voting system is through a cloud based server protected by a control/DMZ server. Each state has its own DMZ server. The voting precincts within each state and their respective ballots are protected by the respective DMZ. Session tickets are required prior to any data exchange. There are two aspects to the voting process: identification and authentication. Biometric and smart card access are used for identification purposes. Authentication occurs during interchange between the smart card and the registration server. In other words, a voter is first identified and then he or she is authenticated prior to voting.

Multilevel authentication. Modular, distributive NFV is critical to this concept. Each of the levels adds a layer of protection against unauthorized access to voter data. Authentication will start with an interaction between the Windows 10 OS and the smart card. Windows 10 OS will interface with the smart card to access the FIDO functionality. Dual identification personal identification number plus fingerprint data are required prior to any access to smart card encrypted data. Authentication between the smart card via the client device and the server will be incorporated at the Transport Layer, using TLS. After security capabilities are determined by the handshake exchange, server authentication and key exchange occurs. The system will use the Web Real-Time Communications (WebRTC) browser. WebRTC is a relatively new secure multimedia browser supported by all major suppliers (e.g. Google, Mozilla, Microsoft). The proof-of-concept will use the Mozilla browser to demonstrate its security features. WebRTC incorporates TLS and HTTPS and adds a layer of security while supporting peer-to-peer interchanges in a multimedia environment (Barnes and Thomson, 2014). WebRTC supports multiple identity systems, including X.509 and OpenID (Barnes and Thomson, 2014). The DMZ functionality is located at the input to the control server. There is also a policy server functionality that directs the outputs depending on the interaction. For example, the control server determines where to send the ballot request packet within the virtual private cloud, depending on the voter information. Most of the concepts associated with each level have been demonstrated. Combining the concepts into a single integrated architecture and layering the identification and authentication processes for Internet voting is unique.

Multilayer encryption. This flexible, extensible architecture offers myriad options to protect voter data with varying levels of security and multifaceted accreditation. Encryption at the network layer, transport layer, and application layer will be used in various scenarios. In addition, hash functions are used for anti-tampering protection in conjunction with Merkle processes and Blockchain structures. Each level of security has its own encryption structure and requirements. Each accreditation request has its own set of protection requirements based on the originator class. Each interface and control is based on the inputs and required outputs of the requesting agents and the database purpose. This layered architecture is portable and can be installed on virtual machines using many of the existing protocols in common use today. However, the proposed architecture will require a few additional protocol structures in order to instantiate the architectural characteristics proposed. Some changes in the management plane will be required for administration, maintenance, health, and status evaluation and reporting. The goal is not to develop an architecture that requires multiple new APIs in order to function. However, the middleware protective software and its relationship to the database information will require some extensive programming and this functionality must be developed, verified, and validated for the proof-of-concept model.

A web-based system can face myriad threats (Stallings, 2014). These threats run the gambit, including integrity, confidentiality, denial of service, man-in-the-middle, and masquerading. Threats can exist at the Web server, Web browser, and network connectivity nodes (Stallings, 2014). The TLS works in conjunction with the Transport Control Protocol (TCP) and provides a secure and reliable end-to-end service. The web client/server interaction also works in conjunction with TLS using HTTPS. TLS provides a peer-to-peer connection relationship. Each connection is associated with a single session (Stallings, 2014). The *Handshake Protocol* is used to create a session between a client and a server. The session defines cryptographic security parameters that can be shared among multiple connections (Stallings, 2014).

In addition to the network aspects, there are also security concerns related to the smart card device and the Windows 10 OS. Smart card access used in this architecture requires authentication. Device authentication in conjunction with biometric identification should offset many of these vulnerabilities included in the threat assessment section. The multilevel authentication includes interactions among the Windows 10 OS, the FIDO Alliance version 2.0 functionality, the smart card, and the control/DMZ server.

IPSec is a security mechanism that prevents unauthorized monitoring and control of network traffic. IPSec operates at the network layer and is compatible with both IPv4 and IPv6 protocols. This means traffic is encrypted at the IP level. IPSec is transparent to users and upper level protocols (Stallings, 2014). In this architecture, tunnel mode is used in authenticated interchanges. Tunnel mode protects the entire IP packet (Stallings, 2014). An encapsulated packet traverses the Internet from a specific point-of-origin to a security gateway. The source and destination addresses can be encapsulated within the packet and protected from external examination. The database server in this architecture will act as a secure gateway and firewall protecting the database addressing infrastructure and data exchange from external entities.

Because this is a thin client architecture, the file server holding the ballot is never transmitted to the user. He or she accesses the ballot during the voting process via the VPN. Prior to accessing the ballot, the system authenticates the user via a Kerberos authentication and access to file server structure. The Kerberos password resides on the smart card and because the smart card requires biometric identification prior to allowing access to its data, there is an added layer of protection. After the Kerberos process, the voter will have access to the appropriate ballot(s) (including multilingual voting) and be able to cast his or her vote over the VPN from his or her computer.

One of the advantages of this thin net architecture is that techniques similar to MixNet (a technique first described by Chaum (1988) and later updated by Chang et al. (2016)) is not required. MixNet is used to obscure client to server packet routing using asymmetric or homomorphic encryption techniques by mixing and re-encrypting packets to obscure input/output order or voter source in routers or servers. This means the data are encrypted and decrypted at several locations along the path to ensure receipt freeness. In the DIVA architecture, receipt freeness is not an issue for two reasons: (1) the thin client structure blocks data required to create a receipt showing voter decisions and (2) because the voter is using a VPN tunnel to transfer the encrypted key strokes and mouse selections to access the ballot through a remote Kerberos-based system, unauthorized access of the plain text data in the channel is highly unlikely. In other words, attacking a VPN tunnel connection would be akin to attacking a single voter; an approach which does not create much benefit versus the cost. The cloud-based thin net Kerberos authenticated interface between the voter and the ballot server is an encrypted simplex connection. This means that while the voter can access the ballot to make his or her selections, the ballot server does not communicate

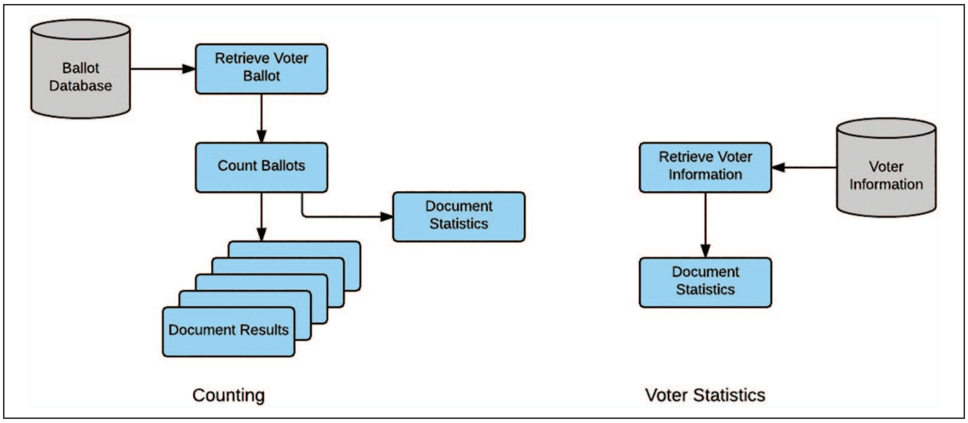


Figure 6. Vote counting process.

with the client system except to provide a display via an encrypted VPN tunnel for the voter to make his or her selections. The client is blocked from downloading or printing the display and the thin client access software disables the print screen function on the client device. In addition to Kerberos timestamps, the confirmation email sent to the voter via separate path includes a timestamp to allow the voter confidence his or her ballot has been submitted without compromising receipt freeness.

Voting integrity. Hash functions are used to protect vote integrity and identify voters who have submitted votes. For example, after a ballot is accepted, a hash function with the voter name, voter ID, trusted ballot, and date and time of vote submission is forwarded to the appropriate database within the virtual private cloud. After voting, each ballot is encrypted using a public asymmetric encryption key associated with the voter's assigned voting precinct This encrypted file is then added to its relevant Merkle tree within the mineblock Blockchain structure using a SHA-256 hash function with nonce key that creates four leading zeros. This process completes a block chain that allows one to quickly identify whether any of the encrypted ballots have been exposed to tampering. If a ballot was tampered with, then the leading zeros would disappear and the chain would be broken, illustrating not only tampering, but also which ballot was altered. The cloud server acknowledges the receipt of the voter's ballot by sending an email (based on voter identification address located in cloud database) and then storing the encrypted ballot and its hash function in another private database. Simultaneously, the hashed voter credentials are stored in a completed voter database. For this architecture the voter credentials are derived from the smart card authentication process and include a hashed biometric signature, the voter registration number in addition to a timestamp that helps thwart any spoofing or man-in-the-middle attacks. These credentials are verified using a SHA-256 hash function to ensure integrity. This process protects the identity of the voter and maintains nonattribution functionality. However, after the ballots are submitted, the hashed voter data can be compared with the precinct voter information database to identify those individuals who voted and preclude them from attempting to vote again in-person. Because the data are not attributable to any particular ballot, nonattribution is maintained. The cloud server then incorporates the voted ballot into a Blockchain structure for each voting precinct using a Merkle tree process. This hashed function is added to the ballot Blockchain within its relevant Merkle tree and forwarded to the vote counting database. This process helps to ensure a voter that his or her vote was accepted and counted without providing any information to a casual observer about how the individual voted. This also protects the ballot from tampering or deletion. Therefore, voter privacy is protected along with assurances that the voter's selections have not been altered or deleted. A database in the cloud holds the hashed sequences and the voter can verify his or her vote was counted by comparing the email value with the database hashed value. Any ballots altered or deleted would be identifiable in the audit using the Blockchain. The proposed Blockchain uses a number used only once (nonce) with four leading zeros and a timestamp. The programming for this type architecture is available open source (Bauer, 2020; Knirsch et al., 2019).

Vote counting. The final process within a voting system architecture is to tally the voter selections and disseminate the results. When a voter submits his or her ballot, the ballot is encrypted using the appropriate voting precinct public key. The private key does not reside within the virtual private cloud. Only a restricted access database at the central vote counting precinct has access to the private key for decrypting the ballot. Because the ballot resides within a closed access blockchain with hash verification, there is confidence that the ballot has not been compromised.

Figure 6 shows an overview of this counting process. The collected ballots are counted in a straightforward manner using slightly modified existing vote counting machines. This is an internal process at a protected precinct location.

Therefore, there are no interactions with external devices involved in vote counting. As part of the vote counting process, statistics on the voters who cast ballots are also captured. Notice that the ballot database and the voter information database are different. In the architecture, the ballot has been encrypted using a precinct public key and the voter information has been hashed at the time of voting and the then the two (voter identification and ballot) are separated into different databases. This precludes voter attribution to a specific ballot. These data were downloaded from the virtual private cloud using a restricted access Internet address and a tunnel VPN. A pdf copy of the ballot can also be printed if there is a need to audit the data.

Concept development

This concept is unique in the fact that I am proposing a *thin client* architecture for this Internet voting scheme. Although a thin client architecture adds network complexity, it also adds security (Pawade et al., 2019) and user simplicity. In this particular architecture, the ballots are not downloaded to the user, but are available for selection via a Kerberos network after authentication. Each state and precinct will have its own particular ballot. One of the advantages of using the thin client approach is that existing ballot software applications can be used and protected. Manufacturers would not need to worry about user device penetration corrupting their ballot program. This adds security for both the voter and the voting infrastructure.

This section describes the research and development of different aspects of the integrated Internet voting architecture. While this proposed architecture is different from previously published options, many of the concepts have existed in the literature for some period of time. For example, Aravind et al. (2019) proposed a concept using both blockchain and fingerprinting for e-voting, while Pawade et al. (2019) also proposed a similar concept, except these authors used iris scanning for their biometric. No biometric system is perfect. For example, if a voter has arch, whorl, or loop damage, then it could be very difficult to obtain a viable fingerprint pattern for recognition. The same could be true for an individual with eye damage. Evaluating these potential deficiencies is beyond the scope of this particular research.

Several of the functions were developed and tested to ensure feasibility prior to integration. The DIVA interactive website is built around Python's Django Framework for the backend code and a Javascript Framework Angular 2. Python was chosen for the project due to it being a universal and powerful language. The Django framework contains a library, Django Rest Framework, making database communication an easier process. In addition, Django allows for integration with Angular 2 to connect information from the website to the Rest API. A MariaDB is incorporated into the architecture. Figure 7 show this framework with its related fields.

Registration process

The url.py file within the Server directory is the main routing file for the website URLs which also contain the child module "Core" URLs. The Core module has a similar URL file that contains the routes to API user endpoint. When this Users endpoint receives a POST request in the browser, the API uses the CreateView in Views.py. This is essentially the controller that handles everything related to an endpoint request and directs the app how to handle the request. The serializer maps all the fields received by the endpoint to our model. The model is what corresponds to our database and contains the field types and validations on the backend side. The Django models create the corresponding tables in the database that is integrated with the application. The various constraints, such as primary key, not null, and unique, are also included in the model. For the DIVA system, this is where a majority of the processing occurs. Figure 4 depicts the process.

The first function call get_voter_id creates a voter ID based on the Voters 2 letter state code and a random UUID4 string. The second and third functions, encryptDOBField and encryptSSNField are calling the encryption functions to pass in the cipher key to our AESCipher class. AES is a symmetric cipher block to encrypt sensitive data. First, our symmetric key is hashed with SHA256 and the block size is set to 32. The key derivation encrypt function within the Django framework will pad the field if needed in accordance with the block size. Next the initialization vector (IV) is generated utilizing Python's crypto library Random function. This is crucial to AES encryption as the IV must be different for each message. Next the field is encrypted and then base64 encoded before sending back to the model. The front-end is primarily divided into three different pages with several components that construct these pages. The first page, the login screen is just temporary for admins to log in for a secure development environment as shown in Figure 8. This page is controlled by the login component which involves authentication. The next page is the Home Page. This is the landing page that the user sees. It gives a basic overview of the DIVA registration process. This page is controlled by the Home component.

The third page is the Registration page, which is the heart of the application. Figure 9 shows this page and the fields that it encompasses.

The form itself includes validation of the information added to the form and will not let the user submit if required fields are not completed. There are also a number of validation functions created on the controller to ensure users can only type numbers in numerical fields. Once the user submits all of the correct validated information, the user service is called which accomplishes a few important tasks prior to calling the API. The documents and images must be encoded in Base64 for storage on Django's Rest Framework API to the database. The File-Upload handles this logic in

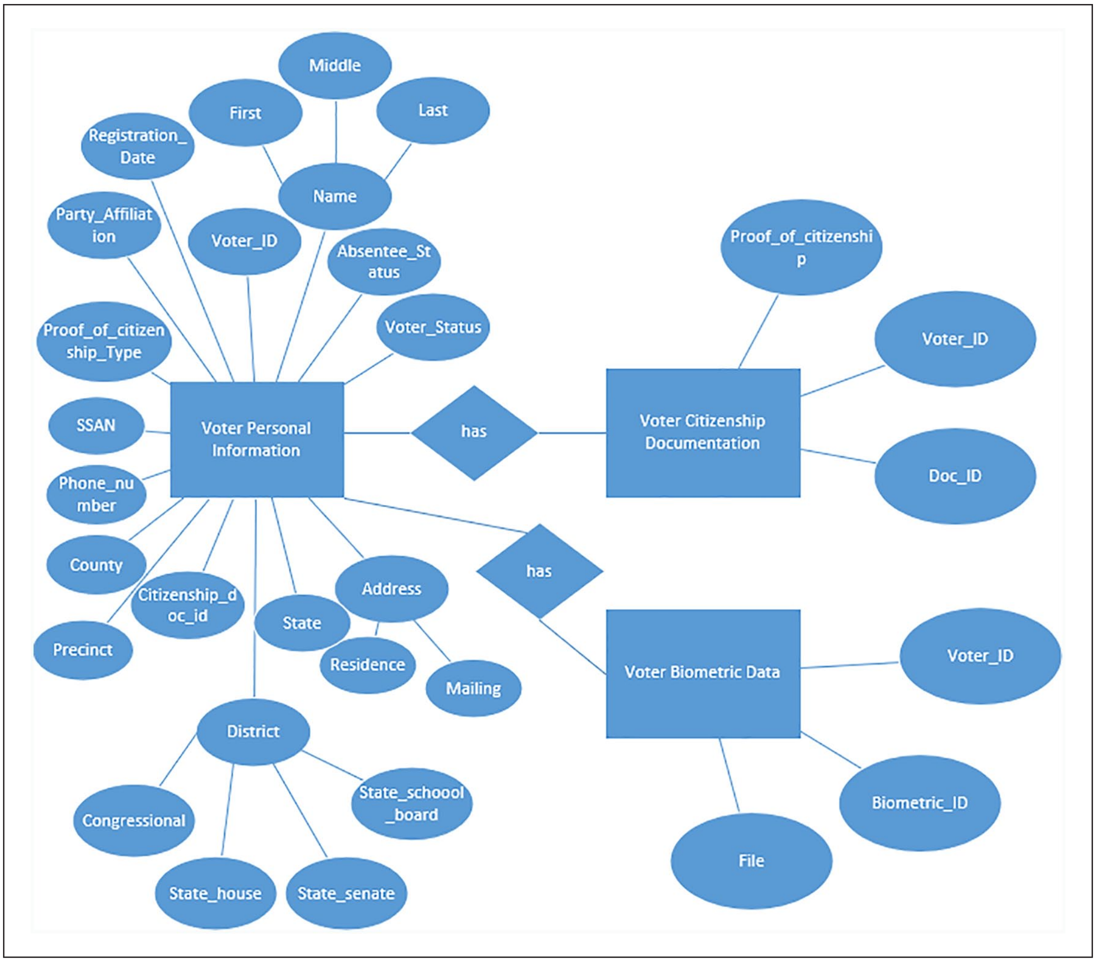
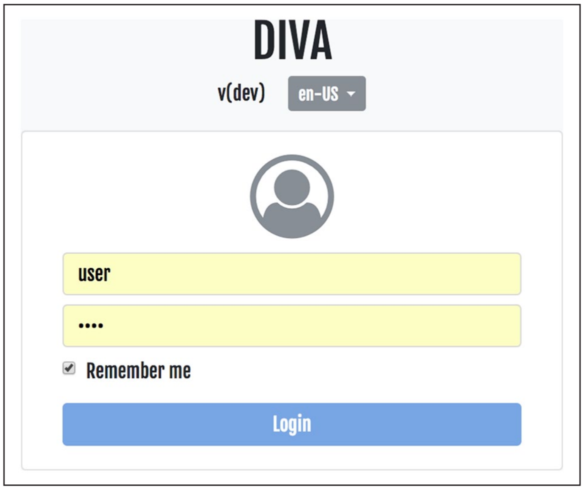


Figure 7. Registration database framework.



user

••••

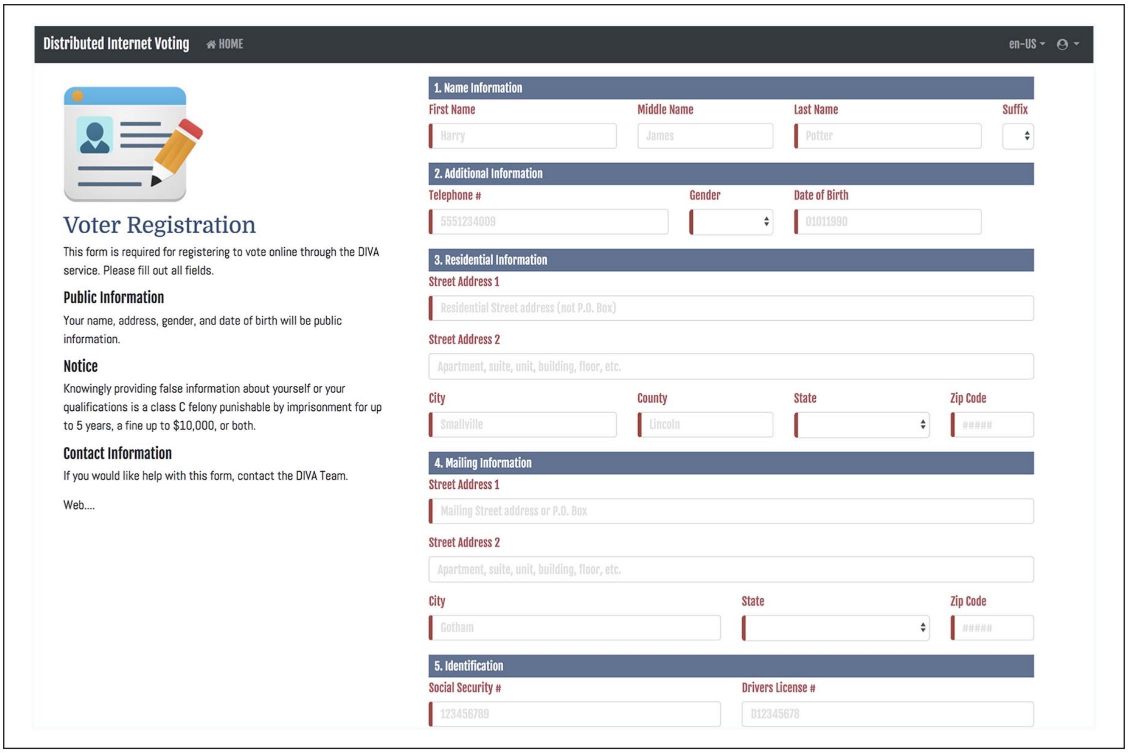
☑ Remember me

Figure 8. Login process.

the readThis function. This function supports the Citizenship field. The User-Service is also another crucial piece of functionality for the front end. It actually adds the documentation to the model after being encoded. It also appends headers, CSRF token, and returns JSON for sending through the POST Request. It also offers Error correction and a redirection to a success page if a 200 Response is received in return.

Figure 10 shows the process required to register the biometric input. Within this proposed architecture, the control server, and database servers will function as the Web Server and the FIDO Server.

For this process, the entire codebase for both the front-end and back-end is hosted using an Apache server on the ASU virtual server infrastructure that was set up for the DIVA project. The MariaDB database resides in the same server as well. A list of tables is autogenerated when the Django application is migrated to the database. Among them, "core_user" table was generated using models.py. However, there was an issue with database migration. Initially, the previous team had used SQLite included Django, later they decided to migrate to MariaDB. Although a guide to connect Django to MariaDB was followed, DIVA was initially connected with SQLite. This was caused by the way Django manages its own database migration file. This problem was corrected by completely removing the SQLite interface. After this issue



Voter Registration

This form is required for registering to vote online through the DIVA service. Please fill out all fields.

Public Information

Your name, address, gender, and date of birth will be public information.

Notice

Knowingly providing false information about yourself or your qualifications is a class C felony punishable by imprisonment for up to 5 years, a fine up to $10,000, or both.

Contact Information

If you would like help with this form, contact the DIVA Team.

Web....

1. Name Information

First Name: Harry
Middle Name: James
Last Name: Potter
Suffix:

2. Additional Information

Telephone #: 5551234009
Gender:
Date of Birth: 01011990

3. Residential Information

Street Address 1: Residential Street address (not P.O. Box)
Street Address 2: Apartment, suite, unit, building, floor, etc.
City: Smallville
County: Lincoln
State:
Zip Code: #####

4. Mailing Information

Street Address 1: Mailing Street address or P.O. Box
Street Address 2: Apartment, suite, unit, building, floor, etc.
City: Gotham
State:
Zip Code: #####

5. Identification

Social Security #: 123456789
Drivers License #: D12345678

Figure 9. Registration page.

was corrected, the database migration files managed by Django were connected to the new database MariaDB.

Biometric information. One of the more difficult challenges was the biometric device hardware that could not be used with any software but the developers' proprietary application. The disparities posed issues. The tools that were purchased from the developer of the biometric device did not seem to work at first and it took some effort to save the scanned image. However, the software issues were resolved and scanning fingerprint images and saving them for upload to the web application was achieved. Currently, the client machine is able to use the software provided with the fingerprint scanner to take a scan of a finger to create a user profile. It then saves this scan as an image for use in uploading to the DIVA application through the form web page. Figure 11 shows the flow of the biometric information as it moves from the fingerprint scanner into the DIVA web application and is combined with the voter ID that was generated based on the user's personal info before being stored on the smart card.

Once the user initiates the finger scan to register for the first time using a biometric device attached to the system, the device captures the image, digitizes the data and stores it in the database in base-64 format. Once the fingerprint and other data are sent to the web application, base-64 formatted data are stored in the database where it can be used for authentication during the voting process as well as generation of the voter identification information stored on the user smart card. This process is followed in the event of a lost smart card by an individual. This process has been successfully implemented and verified.

Smart cards. The smart cards are able to be read and written to using open source software available on Red Hat Linux. Red Hat Linux provides all of the necessary tools for the creation and use of smart cards. The selected approach uses a contact smart card with a smart chip (i.e. microprocessor) embedded into the card. A smart card reader is required. The architecture uses a CPU/MPU Microprocessor Multi-function Card. In addition to fingerprint data, the card also supports public key infrastructure (PKI) public key code associated with the precinct voting site. The card has an on-board math coprocessor. I recognize that cost could be a drawback; however, some of these costs could be offset by needing fewer personnel for counting and sorting of mail-in ballots. Over time, this could result in a more cost-effective approach to voting remotely for those who cannot feasibly vote at a certified polling place.

The smart card is programmed with data that include voter identification information; public key information for the user voting precinct; secret cookie for establishing initial interconnectivity via FIDO; restricted access IPv6 address for VPN tunnel establishment. As part of the layered security, a separate dual authentication process is required for thin client access to the ballot. For example, the user will have access to a digitally signed ballot in the thin client server so that he or she has confidence that the ballot they are submitting is not an imitation or counterfeit.

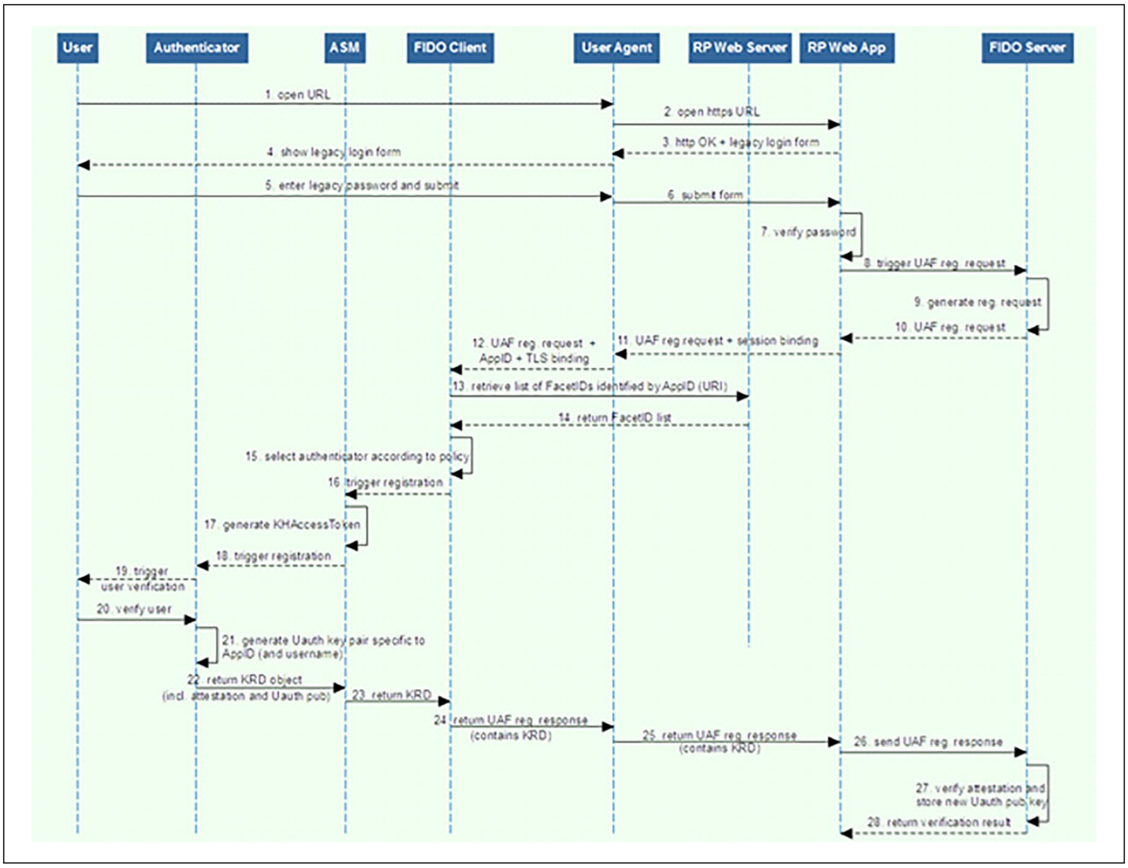

User
Authenticator
ASM
FIDO Client
User Agent
RP Web Server
RP Web App
FIDO Server
1. open URL
2. open https URL
3. http OK + legacy login form
4. show legacy login form
5. enter legacy password and submit
6. submit form
7. verify password
8. trigger UAF reg. request
9. generate reg. request
10. UAF reg. request
11. UAF reg request + session binding
12. UAF reg. request + AppID + TLS binding
13. retrieve list of FacetIDs identified by AppID (URI)
14. return FacetID list
15. select authenticator according to policy
16. trigger registration
17. generate KHAccessToken
18. trigger registration
19. trigger user verification
20. verify user
21. generate Uauth key pair specific to AppID (and username)
22. return KRD object (incl. attestation and Uauth pub)
23. return KRD
24. return UAF reg. response (contains KRD)
25. return UAF reg. response (contains KRD)
26. send UAF reg. response
27. verify attestation and store new Uauth pub key
28. return verification result

Figure 10. Biometric registration process.

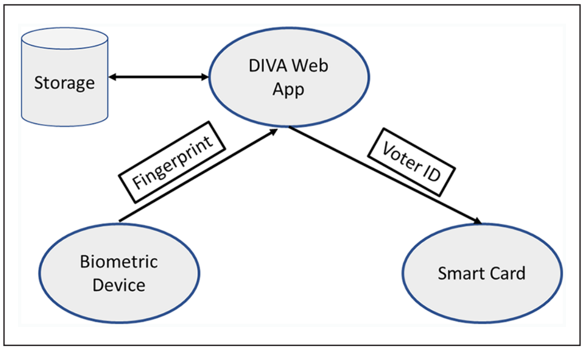


Figure 11. Model of the flow of biometric information to Smart Card.

The asymmetric public key is used during smart card dual authentication to validate the precinct digital signature associated with the ballot for the voter. The biometric data are used by the VPC servers in the system to support authentication of the voter. The smart card interacts with the VPC ballot server via its VPN tunnel link and establishes a thin client connection with its precinct ballot server. Prior to connecting to the ballot, a Kerberos process authenticates and then grants ticket access to the voter. The voter then directly accesses the ballot for his or her precinct. Smart card programming and reading of data has been successfully implemented and verified.

Voting process

The voting process begins with two-factor authentication. After authentication is complete, the voting process begins. The VPN Gateway's IP address is a function of user location and the interactions with the myriad of virtual private cloud servers and databases. The IP address is a function of the particular state and voting precinct associated with the voter. Figure 12 shows the ICT flow for the thin client process. However, due to the overall complexity of the system, not all functions can be shown in detail. However, Figure 12 does provide a good overview of the system connectivity and several aspects of this process have been tested. Almost all of the cloud-based functionality, including ballot encryption and blockchain functions and VPN ICT have been developed and tested.

Multilevel authentication. Modular, distributive NFV is critical to this concept. Each of the levels adds a layer of protection against unauthorized access to voter data. Authentication will start with an interaction between the Windows 10 OS and the smart card. Windows 10 OS will interface with the

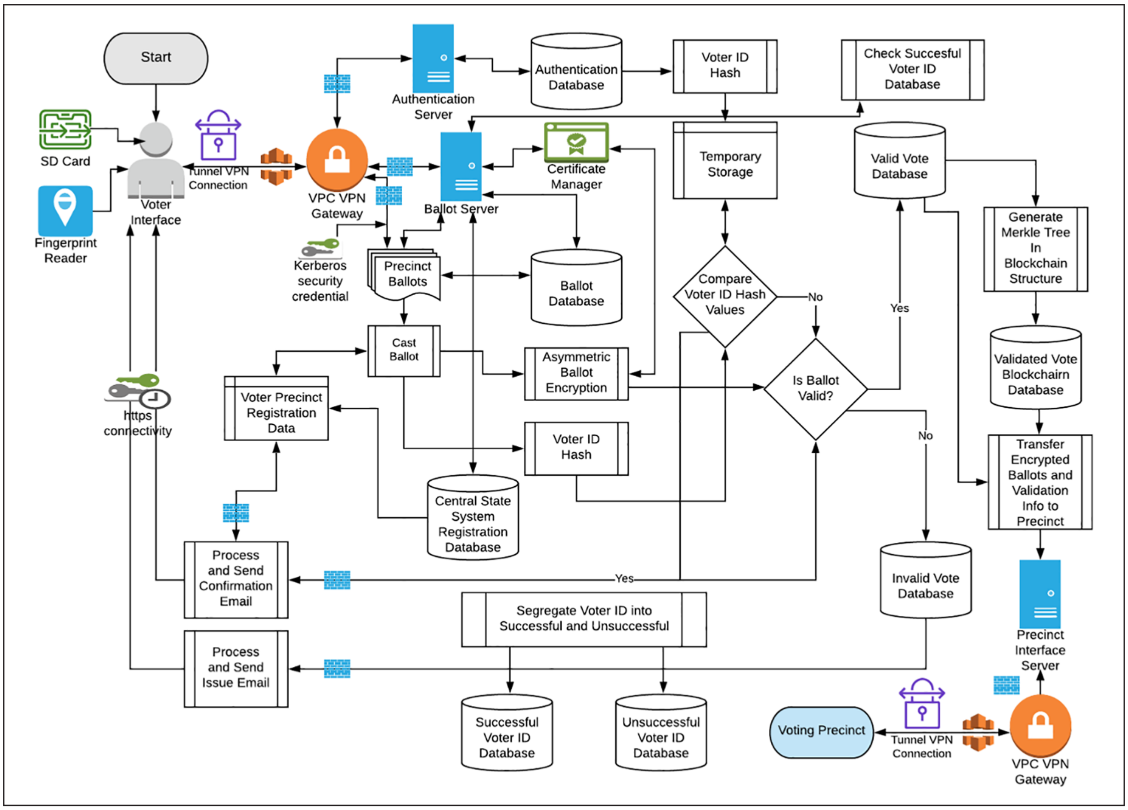

Start
SD Card
Fingerprint Reader
Voter Interface
Tunnel VPN Connection
VPC VPN Gateway
Authentication Server
Authentication Database
Voter ID Hash
Check Succesful Voter ID Database
Ballot Server
Certificate Manager
Temporary Storage
Valid Vote Database
Generate Merkle Tree In Blockchain Structure
Kerberos security credential
Precinct Ballots
Ballot Database
Compare Voter ID Hash Values
No
Yes
Is Ballot Valid?
Validated Vote Blockchairn Database
Cast Ballot
Asymmetric Ballot Encryption
Voter Precinct Registration Data
https connectivity
Voter ID Hash
No
Transfer Encrypted Ballots and Validation Info to Precinct
Central State System Registration Database
Process and Send Confirmation Email
Yes
Invalid Vote Database
Precinct Interface Server
Segregate Voter ID into Successful and Unsuccessful
Process and Send Issue Email
Successful Voter ID Database
Unsuccessful Voter ID Database
Voting Precinct
Tunnel VPN Connection
VPC VPN Gateway

Figure 12. Thin client ICT voting process overview.

smart card to access the FIDO functionality. Dual identification personal identification number plus fingerprint data will be required prior to any access to smart card encrypted data. Authentication between the smart card via the client device and the server will be incorporated at the Transport Layer, using TLS and a VPN Gateway whose IP location exists in the smart card. After security capabilities are determined by the handshake exchange, server authentication and key exchange occurs. The system will use the Web Real-Time Communications (WebRTC) browser. WebRTC is a relatively new secure multimedia browser supported by all major suppliers (e.g. Google, Mozilla, Microsoft). The proof-of-concept uses the Mozilla browser to demonstrate its security features. WebRTC incorporates TLS and HTTPS and adds a layer of security while supporting peer-to-peer interchanges in a multimedia environment (Barnes and Thomson, 2014). WebRTC supports multiple identity systems, including X.509 and OpenID (Barnes and Thomson, 2014). The DMZ functionality is located at the input to the virtual private cloud gateway server. There is also a policy server functionality that directs the outputs depending on the interaction. For example, the control server determines where to send the ballot request packet, depending on the voter information. Most of the concepts associated with each level have been demonstrated. Combining the concepts into a single integrated architecture and layering the identification and authentication processes for Internet voting is unique.

Databases and servers. The architecture resides within a virtual private cloud environment. Virtual servers are distributed. The DMZ architecture with network address translation and IPv6 addressing protects infrastructure by limiting client access to IP addressing information. The client interacts with the VPN Gateway. A virtual control/DMZ server completes the TLS Handshake process. The TLS type process helps accomplish authentication. During the authentication process, requests and cookies will be exchanged among the client, the control/DMZ server, and other virtual servers supporting the infrastructure. A secret control cookie for authentication is programmed into the smart card. An internal domain name system (DNS) virtual server interacts with the virtual file transfer server to determine the correct IP address for the authenticated response. During the voting process, the virtual file transfer server transfers the data based on the IP address associated with the .

Database access and security will be protected through middleware software. This gateway keeper software will prevent unauthorized access to the databases and protect sensitive empirical data. In addition, extremely sensitive empirical data within the database will be encrypted prior to being stored. As discussed earlier, a Kerberos ticket granting process will be used between the voter and the ballot casting server during his or her voting process. This adds to the integrity of the voting system.

Multilayer encryption. This flexible, extensible architecture offers myriad options to protect voter data with varying levels of security and multifaceted accreditation. Encryption at the network layer, transport layer, and application layer is used. Each level of security has its own encryption structure and requirements. Each ballot access request has its own set of protection requirements based on the originator data. Each interface and control is based on the inputs and required outputs of the requesting agents and the database purpose. This layered architecture is portable and can be installed on virtual machines using many of the existing protocols in common use today. However, the proposed architecture will require a few additional protocol structures in order to instantiate the architectural characteristics proposed. Some changes in the management plane will be required for administration, maintenance, health, and status evaluation and reporting. The goal is not to develop an architecture that requires multiple new APIs in order to function. However, the middleware protective software and its relationship to the database information will require some extensive programming and this functionality must be developed, verified, and validated for the integrated proof-of-concept model.

A web-based system can face myriad threats (Stallings, 2014). These threats run the gambit, including integrity, confidentiality, denial of service, man-in-the-middle, and masquerading. Threats can exist at the Web server, Web browser, and network connectivity nodes (Stallings, 2014). The TLS works in conjunction with the TCP and provides a secure and reliable end-to-end service. The web client/server interaction also works in conjunction with TLS using HTTPS. TLS provides a peer-to-peer connection relationship. Each connection is associated with a single session (Stallings, 2014). The *Handshake Protocol* is used to create a session between a client and a server. The session defines cryptographic security parameters that can be shared among multiple connections (Stallings, 2014).

In addition to the network aspects, there are also security concerns related to the smart card device and the Windows 10 OS. Smart cards used in this architecture will be encrypted. Device authentication in conjunction with biometric identification should offset many of these vulnerabilities. The multilevel authentication includes interactions among the Windows 10 OS, the FIDO Alliance version 2.0 functionality, the smart card, and the control/DMZ server.

IPSec is a security mechanism that prevents unauthorized monitoring and control of network traffic. IPSec operates at the network layer and is compatible with both IPv4 and IPv6 protocols. This means traffic is encrypted at the IP level. IPSec is transparent to users and upper level protocols (Stallings, 2014). In this architecture, tunnel mode is used in authenticated interchanges. Tunnel mode protects the entire IP packet (Stallings, 2014). An encapsulated packet traverses the Internet from a specific point-of-origin to a security gateway. The source and destination addresses can be encapsulated within the packet and protected from external examination. The virtual cloud gateway in this architecture will act as a secure gateway with firewalls protecting the cloud addressing infrastructure and data exchange from external entities.

Voting integrity. Symmetric encryption is used in the ICT interactions between the voter client and the virtual cloud gateway. Asymmetric encryption is used for ballot authenticity verification and for ballot protection after the voter has completed his or her ballot. Hash functions are also used to help with user and ballot validation. Merkle trees have been used in cloud based systems to help segregate and secure data (Mao et al., 2015). A blockchain structure is also used to assist in the validation of a series of ballots within a particular voting precinct. The asymmetrically encrypted ballots are hashed and then chained by using the hash of a previous block. In other words, each block will store the hash of the previous block. These blocks are chained using a nonce with four leading zeros. The advantage of this blockchain process is that if a particular ballot has been modified in any manner, the hash chain is broken and it is easy to locate which ballot has experienced some type of tampering. If there are no indications of tampering, then one can be confident that all of the ballots are in their original form. The private key required to decrypt the ballots prior to counting resides only at the respective precinct and is never transmitted across the network.

Hash functions will also be used to identify voters who have submitted votes. For example, after a ballot is accepted, a hash function with the voter name, voter ID, trusted ballot, and date and time of vote submission is forwarded to a cloud server. The cloud server acknowledges the receipt of the voter’s ballot, storing the ballot hash function in a privately accessible database. The cloud server then incorporates the voted ballot into a Merkle tree. This hashed function is added to the ballot and forwarded to the vote counting database that resides offline at the relevant precinct. The hashed voter ID is also verified. An email response to the voter indicating whether or not his or her ballot was accepted is generated. The response incorporates TLS which is based on Diffie-Hellman (Lake, 2019). This also protects the ballot from tampering or deletion. Therefore, voter privacy is protected along with assurances that the voter’s selections have not been altered or deleted. A database in the cloud holds the hashed sequences until they are downloaded to the precinct. The voter can verify his or her vote was counted using the information found in the email. Any ballots altered or deleted would be identifiable in the audit using this approach.

Vote counting

The final process within a voting system architecture is to tally the voter selections and disseminate the results. Voter completed ballots will be locked when a voter submits his or her vote. In other words, no changes can be made to a ballot after the voter submits his or her choices. To prevent tampering the ballots are immediately encrypted at the origination point using asymmetric encryption. These ballots will eventually be forwarded to the system owner where they can only be decrypted by a private key. The private key is kept in an offline system. Votes are downloaded from the VPC to the precinct and then transferred via a secure USB to the closed network that conducts vote counting. The public encryption key supporting any particular election resides on the smart card and in the virtual private cloud as a part of the voter authentication process. The collected ballots are counted in a straightforward manner. This is an internal process. Therefore, there are no interactions with external devices involved in vote counting. As part of the vote counting process, statistics on the voters who cast ballots are also captured. Notice that the ballot database and the voter information database are different. This precludes voter attribution to a specific ballot. Figure 6 shows the vote counting process. Vote counting requires the least amount of concept design because there are several existing machines that could be used for this process.

The ballots and supporting data are downloaded from the virtual private cloud using a limited access IP address within a VPN tunnel. The ballots are counted electronically after decryption and the results transferred to a secure encrypted database. The ballots cannot be decrypted prior to arrival at the system owner vote counting server. The ballots are also stored in encrypted form in a database for purposes of recount and historical evidence, as well as, auditing purposes. The Merkle tree blockchain hash functions are also stored. This process prevents tampering because while an administrator or any individual who gained access could view and process a vote, he or she could not alter or delete the ballot data without being discovered because an audit which re-encrypts the ballot would result in a different hash function for the altered voted ballot.

Findings

The primary research question for this effort is: "How can the concept of a cybersecure Internet voting process that incorporates voter integrity, nonrepudiation, voter privacy, and accurate vote recording with voter nonattribution be architected?", Figures 7, 11, and 12 show how this thin client architecture supports these hypotheses.

H1 results. Two-factor authentication including biometric identification was demonstrated as a viable security consideration. It was also demonstrated that modern browser technologies can be used to enhance privacy and security of this approach. There are multiple security layers in the architecture. A limited access VPN interacting with a VPC protects ICT interactions. In addition to this IPSec cryptographic protection, the interactions are also protected inside the envelope using transport layer security (TLS). Kerberos is used as a security solution to the thin client interaction between the voter's client computer and the VPC Ballot. Each of these individual concepts have been shown to work. The concepts developed to test this hypothesis showed that a multilevel, multilayer, distributed virtual system using modified browser technologies and modern ICT could support the integrated architecture.

H2 results. Critical information is stored on smart cards which cannot be accessed without proper credentials and the contents cannot be modified. Fingerprint identification directly supports unauthorized access and nonrepudiation. The precinct has confidence that the individual who accessed and voted is the registered individual. The fingerprint acts a digital signature. Both the smart card and fingerprint concepts were demonstrated. Unauthorized access and ballot modification are also mitigated by the use of Merkle trees in a blockchain architecture. The encrypted ballots are protected from unidentified tampering through the use of blockchain connectivity with a nonce. This concept was demonstrated.

Vote recording uses a thin client interaction between the voter and the ballot file. This Kerberos protected interaction allows the voter to make his or her selections and check results prior to submission. When the ballot is completed, it is encrypted with his or her precinct public key. The ballot is not decrypted until counting. The header information is structured in a hash function to protect voter identity. These steps mitigate any voter attribution and support the nonattribution requirement. The tunnel VPN between the VPC and the voter client computer mitigate man-in-the-middle attacks and denial-of-service (DoS) attacks. Also, the structure that incorporates multiple VPN gateways for access to the VPC mitigates the opportunity for any type DoS attack. Because data are stored on a smart card and most important values are verified using hash functions, the opportunity to insert malware onto the voter device that modifies any key characteristics is highly unlikely. The concepts developed to test this hypothesis showed that mitigating unauthorized access, man-in-the-middle attacks, malware, and denial-of-service attacks could support the integrated architecture.

H3 results. The unique nature of this concept uses a modularized thin client architecture. The voter registration concept is critical to the overall success of this Internet voting concept. Key components of this first process are the collection of fingerprint data during the registration process. The fingerprint data are used in two-factor authentication and other identification procedures during the voting process. The modular technology includes segregation of functionalities in the voter client hardware and the VPC. Within the voter client hardware, the smart card segregates critical information from the client computer and blocks access to unauthorized users. The fingerprint process provides one aspect of the two-factor authentication. The VPC functionality is also modularized. Databases and processes are segregated. The voter identification characteristics and ballot choices are separated to ensure voter privacy. The Merkle tree process with blockchain structure helps preclude any vote tampering; therefore, the integrity of the process is protected. Various modules of the voting process were developed and tested to show that a voting system cyberinfrastructure can be implemented using existing technology. However, some software development to support various interfaces and specific activities was required.

Applications

This proposed architecture is targeted at two particular groups: military and other individuals supporting the US government roles who are remote and away from their normal voting precincts; and, senior citizens, particularly those senior citizens who are residing in assisted living facilities and lack mobility. Each of these groups could benefit from this approach. With the recent concerns about fraud in the mail-in balloting process, this approach mitigates most of those concerns.

Military and foreign service

One advantage in using this approach for military voting is that the military currently use a smart card identification program called Common Access Card (CAC). The CAC became the standard identification and access card for the Department of Defense in the early 2000s (Ardiley, 2012). The CAC card stores fingerprint data, PKI certificates, personal identification verification certificate, plus other attributes. This means that an Internet voting approach that uses a smart card and fingerprint techniques would not be a new experience for these personnel.

Senior citizens

Senior citizens who live in assisted living do not have the same experiential knowledge about smart cards and fingerprint identification as those in the military. However, research has shown that the majority these senior citizens would be willing to learn in order to vote in an election (Helm, 2015). The key to implementing this program and ensuring its success would be to offer some training. This training could either be accomplished online or in-person. The advantage could be that it may well mitigate nonparticipation by this normally active voting bloc. This means that residing in assisted living could be less likely to contribute to a disenfranchising culture of senior citizens living in these closed communities.

Conclusion

The development and demonstration of the concepts associated with this architecture show that the theoretical hypotheses have merit. The programmed and executed architectural concepts have demonstrated the cogency of many aspects associated with an integrated secure Internet voting approach. Within this context, the security features developed as described in *Objectives* 1–4 were an integral part of the design and implementation model. Database access and security is protected through middleware software. This gateway keeper software in the tunnel VPN structure prevents unauthorized access to the VPC databases and servers and protects sensitive empirical data. Any database modification by administrators requires two-person verification. In addition, extremely sensitive data within the database are encrypted prior to being stored (e.g. SSANs or ballots). Only internal administrative users will have access to sensitive information within any database. Layered encryption, tunnel VPN techniques, biometric data, multilevel authentication within a thin client virtual environment all contribute to the security and privacy of this approach. The concept demonstrations have shown the feasibility. However, before this system could be deployed, an integrated system would have to created and subjected to penetration testing.

It should be emphasized that although the conceptional development and demonstrations were successful and therefore the proposed concept appears feasible, there are issues that remain. For example, a completely integrated system that includes manufacturer equipment still needs to be developed. Scaling could also be a factor. Although most cloud architectures are elastic, it still needs to be determined whether there are any scaling factors. The conceptual demonstrations did not show a validated end-to-end system. This would be required in conjunction with the penetration testing.

One other issue must be identified, even though it is beyond the scope of this technical research to resolve. One of the biggest hurdles to overcome could be political. The US voting structure delegates most of the voting approaches to individual states (Issacharoff, 2015). This means that how a state decides to vote in any general election is a decision of that particular state. In other words, developing a ubiquitous Internet voting structure for military and senior citizens could be time-consuming and problematic.

In summary, this research has shown that recent technological advances in network security have advanced an Internet voting initiative and demonstrated that a secure Internet voting approach is feasible. Although it is nearly impossible to design a perfect system, I believe this system creates enough roadblocks to mitigate undesired intrusions into either the voter device or the VPC. Keep in mind that the goal is not perfection. The goal is to make unauthorized access extremely difficult. In any security environment, one of the goals is to evaluate the cost/benefit ratio between the effort to attack the system versus the reward. A distributed system has an inherent advantage in that even if an attack is successful, it is a limited assault. In addition, voting windows for elections are relatively short-lived. The random nature of when any particular voter would participate coupled with the identified ICT and authentication security results in a very secure Internet voting system architecture.

Acknowledgements

The concept demonstration of this conceptual architecture would have been more difficult and taken more time if not for the help of several Arizona State University (ASU) students. These students assisted by programming scripts and executing test scenarios that demonstrated the feasibility of various concepts described in this research. Each contributed, to varying degrees at different times in the development during the past 2 years. Normally each student worked on a particular aspect for 2–3 months. In no particular order, these individuals included Tiffany Denkler, Apoorva Telukuntla, Hyunwook Park, Ruchira Dorsatwar, Brandon Gatto, Zachary Dooley, Christopher Knuth, and Ian Jacobs. Their assistance helped shape the architectural concept and demonstrate the feasibility of using existing technology to create a cybersecure Internet voting system.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Jim E Helm https://orcid.org/0000-0002-5184-7004

Supplemental material

Supplemental material for this article is available online. Contact jim.helm@asu.edu for access.

References

Adida B (2008) Helios: Web-based open-audit voting. In: *17th USENIX security symposium* (ed PC Van Oorschot), San Jose, CA, 28 July–1 August, pp. 335–348. USENIX Association.

Ahmad AA, Alajeely M and Doss R (2016) Establishing trust relationships in OppNets using Merkle trees. In: *2016 8th international conference on communication systems and networks (COMSNETS)*, Bangalore, India, 5–10 January.

Al-Ameen A and Talab S (2013) The technical feasibility and security of e-voting. *International Arab Journal of Information Technology (IAJIT)* 10: 397–404.

Al-Anie HK, Alia MA and Hnaif AA (2011) e-Voting protocol based on public-key cryptography. *International Journal of Network Security and Its Applications* 3: 87–98.

Alvarez RM and Hall TE (2004) *Point, Click, and Vote: The Future of Internet Voting*. Washington, DC: Brookings Institution Press.

Alvarez RM and Hall TE (2008) Building secure and transparent elections through standard operating procedures. *Public Administration Review* 68: 828–838.

Anooja A (2016) Internet voting system and digital India. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)* 5: 62–64.

Aravind P, Gokul RS, Mohanraj S, et al. (2019) Blockchain and finger print enabled E-Voting. *SSRG International Journal of Electronics and Communication Engineering (SSRG-IJECE)* 6: 8–13.

Ardiley S (2012) *History of the Common Access Card (CAC)*. Security Infowatch. Available at: https://www.securityinfowatch.com/home/article/10653434/history-of-the-common-access-card-cac (accessed 27 May 2020).

Awad M (2011) *Using cryptography and enhanced verification to safeguard electronic voting*. PhD Thesis, University of Houston, Houston, TX.

Bailey S, Bansal D, Dunbar L, et al. (2012) *Software-Defined Networking: The New Norm for Networks*. Palo Alto, CA, USA Open Networking Foundation.

Bakker A, Petrocco R and Grishchenko V (2015) Peer-to-peer streaming peer protocol (PPSPP). *Internet Engineering Task Force (IETF)*.

Barnes RL and Thomson M (2014) Browser-to-browser security assurances for WebRTC. *IEEE Internet Computing* 18: 11–17.

Bauer G (2020) Programming blockchains step-by-step from scratch (Zero). Starting with crypto hashes. In: Moto YA (ed.). Yuki and Moto.

Bélanger F and Carter L (2012) Digitizing government interactions with constituents: An historical review of e-government research in information systems. *Journal of the Association for Information Systems* 13: 363–394.

Beroggi GEG (2008) Secure and easy Internet voting. *IEEE Computer Magazine* 41: 52–56.

Bharadwaj V, Le Van Gong H, Balfanz D, et al. (2016) *Web Authentication: An API for Accessing Scoped Credentials*. Available at: https://www.w3.org/TR/webauthn/ (accessed 30 September 2016).

Buckland R, Teague V and Wen R (2012) Towards best practice for E-election systems. In: Kiayias A and Lipmaaa H (eds) *Lecture Notes in Computer Science*. Berlin: Springer, pp. 224–241.

Carter LD (2006) *Political participation in a digital age: An integrated perspective on the impacts of the internet on voter turnout*, Doctoral dissertation, Virginia Polytechnic Institute and State University, Blacksburg, VA.

Carter LD and Bélanger F (2005) The utilization of e-government services: Citizen trust, innovation and acceptance factors. *Information Systems Journal* 15: 5–25.

Carter LD and Campbell R (2011) The impact of trust and relative advantage on Internet voting diffusion. *Journal of Theoretical and Applied Electronic Commerce Research* 6: 28–42.

Carter LD, Schaupp LC, Hobbs J, et al. (2011) The role of security and trust in the adoption of online tax filing. *Transforming Government: People, Process and Policy* 5: 303–318.

Chang D, Chauhan AK, Kang J, et al. (2016) Apollo: End-to-end verifiable voting protocol using mixnet and hidden tweaks.

In: Kwon S and Yun A (eds) *Lecture Notes in Computer Science*. Cham: Springer, pp. 194–209.

Chaum E (1988) Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA. In: Barstrow D (ed.) *Lecture Notes in Computer Science*. Berlin: Springer, pp. 177–182.

Chevallier M (2009) Internet voting, turnout and deliberation: A study. *Electronic Journal of E-Government* 7: 29–44.

Chipman I (2016) *David Dill: Why Online Voting Is a Danger to Democracy*. Stanford Engineering. Available at: https://engineering.stanford.edu/magazine/article/david-dill-why-online-voting-danger-democracy (accessed 21 May 2020).

Chung KK-T, Minadeo JA and Shi X (2005) *Packet-based Internet Voting Transactions with Biometric Authentication*. U. S. Patent application 10/348, 433.

Clarkson MR, Chong S and Myers AC (2008) Civitas: Toward a secure voting system. In: *2008 IEEE Symposium on Security and Privacy,* Evans, D. (ed), Oakland, CA, USA: IEEE Computer Society. IEEE Computer Society.

Cortes E and Norden L (2018) *Paper Trails for All*. Slate. Available at: https://slate.com/technology/2018/11/paperless-voting-machines-upgrades-russia-2020-election-security.html (accessed 22 May 2020).

Davide B, Greg B, Marco C, et al. (2010) An experience in testing the security of real-world electronic voting systems. *IEEE Transactions on Software Engineering* 36: 453–473.

Davis FD (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly* 13: 319–340.

Dill DL and Castro D (2008) The U.S. should ban paperless electronic voting machines. *Communications of the ACM* 51: 29–33.

Django (2020) *Archive of Security Issues*. Available at: https://docs.djangoproject.com/en/3.0/releases/security/ (accessed 20 June 2020).

Eom T, Hong JB, Wang G, et al. (2015) Security modeling analysis of a based web service. In: *ICA3PP Internation Workshops and Symposiums* (Wang, G., Zomaya, A., Perez, G. M. & Li, K., eds.), 18-20 November 2015, pp. 746-756. Zhangjiajie, China: Springer International Publishing, pp. 746–756.

Epstein J (2013) Are all types of internet voting unsafe? *IEEE Security & Privacy* 11: 3–4.

FIDO Alliance (2014) *FIDO Application Identification and Facet*. FIDO Alliance. Available at: https://fidoalliance.org/specifications/overview/ (accessed 28 March 2016).

File T (2018) Characteristics of voters in the presidential election of 2016. In: United States Census Bureau (ed.) *Current Population Survey Reports*. Washington, DC: United States Census Bureau, pp. 1–20.

Fink RA (2010) *Applying trustworthy computing to end-to-end electronic voting*. PhD Thesis, Department of CSEE, University of Maryland, Baltimore County, Baltimore, MD.

FVAP (2017) *Post election Voting Surveys Active Duty Military: Technical Report*. Washington, DC: Marsh Group, LLC.

Gerlach F (2009) Seven principles for secure e-voting. *Communications of the ACM* 52: 8.

Germann M, Micha G and Uwe S (2016) Internet voting for expatriates: The Swiss case. *Ejournal of Edemocracy and Open Government* 6: 197–215.

Gibson JP, Krimmer R, Teague V, et al (2016) A review of E-voting: The past, present, and future. *Annals of Telecommunications* 71: 279–286.

Halderman JA and Teague V (2015) The New South Wales iVote system: Security failures and verification flaws in a live online election. In: Haenni R, Koenig RE and Wikstrom D (eds) *Lecture Notes in Computer Science*. Cham: Springer, pp. 35–53.

Haleplidis E, Pentikousis K, Denazis S, et al. (2015) Software-defined networking (SDN): Layers and architecture terminology. *Internet Research Task Force (IRTF)*, RFC 7426.

Helm JE (2015) Internet e-Voting: How technology acceptance and the digital divide influence senior citizen intention to use a new voting technology. UMI, Proquest LLC.

Howlader J, Nair V, Basu S, et al. (2011) Uncoercibility in e-voting and e-auctioning mechanisms using deniable encryption. *International Journal of Network Security and Its Applications* 3: 97–109.

Issacharoff S (2015) Ballot bedlam. *Duke Law Journal* 64: 1363–1409.

Jefferson D, Rubin AD, Simons B, et al. (2004a) *A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE)*. ACM. Available at: http://usacm.acm.org/evoting/category.cfm?cat=14&E-Voting (accessed 15 July 2011).

Jefferson D, Rubin AD, Simons B, et al. (2004b) Analyzing Internet voting security. *Communications of the ACM* 47: 59–64.

Jin J, Jones MB and Lindemann R (2015) *Web API for Accessing FIDO 2.0 Credentials*. Available at: https://www.w3.org/Submission/2015/SUBM-fido-web-api-20151120/#dependencies (accessed 28 March 2016).

Kaliyamurthie KP, Udayakumar R, Parameswari D, et al. (2013) Highly secured online voting system over network. *Indian Journal of Science and Technology* 6: 4831–4836.

Kavakli E and Gritzalis S (2007) Protecting privacy in system design: The electronic voting case. *Transforming Government: People, Process and Policy* 1: 307–307.

Knirsch F, Unterweger A and Engel D (2019) Implementing a blockchain from scratch: Why, how, and what we learned. *EURASIP Journal on Information Security* 2. DOI: 10.1186/s13635-019-0085.

Lake J (2019) *What Is the Diffie-Hellman Key Exchange and How Does It Work?* Available at: https://www.comparitech.com/blog/information-security/diffie-hellman-key-exchange/ (accessed 26 May 2020).

Mao J, Zhang Y, Li P, et al. (2015) A position-aware Merkle tree for dynamic cloud data integrity verification. *Soft Computing* 19: 1–14.

Meko T, Lu D and Gamio L (2016) How Trump won the presidency with razor-thin margins in swing states. *The Washington Post*, November 16. Available at: https://www.washingtonpost.com/graphics/politics/2016-election/swing-state-margins/ (accessed 16 November 2020).

Meng B and Wang JQ (2010) An efficient receiver deniable encryption scheme and its applications. *Journal of Networks* 5: 683–690.

Mohammadpourfard M, Doostari MA, Ghaznavi Ghoushchi MB, et al. (2015) A new secure Internet voting protocol using Java Card 3 technology and Java information flow concept. *Security and Communication Networks* 8: 261–283.

Monthly L (2019) *5 Technological Advancements that Are Boosting Cybersecurity*. Available at: https://www.lawyer-monthly.com/2019/02/5-technological-advancements-that-are-boosting-cybersecurity/ (accessed 22 May 2020).

Neumann S, Volkamer M, Budurushi J, et al. (2016) SecIVo: A quantitative security evaluation framework for internet voting schemes. *Annals of Telecommunications* 71: 337–352.

Parks M (2019) *In 2020, Some Americans Will Vote on Their Phones. Is That the Future?* Available at: https://www.npr.org/2019/11/07/776403310/in-2020-some-americans-will-vote-on-their-phones-is-that-the-future: NPR. Available: https://www.npr.org/2019/11/07/776403310/in-2020-some-americans-will-vote-on-their-phones-is-that-the-future (accessed 17 May 2020).

Pawade D, Sakhapara A, Badgujar A, et al (2019) Secure online voting system using biometric and blockchain. In: Sharma N, Chakrabarti A and Balas VE (eds) *ICDMAI 2019, Volume I Data Management, Analytics and Innovation*. Kuala Lumpur, Malaysia: Springer, pp. 93–110.

Rogers EM (2003) *Diffusion of Innovations*, *[E-pub Reader Version]* (5th ed.). New York: Free Press.

Rosacker RE and Rosacker K (2012) A call for collaborative academic and practitioner efforts to address remote-access voting methods. *Transforming Government: People, Process and Policy* 6: 230–238.

Sabin ZP (2018) *Why Electronic Voting Is a Bad Idea*. Medium. Available at: https://medium.com/@zacharysabin/why-electronic-voting-is-a-bad-idea-7bdedea2bce1 (accessed 21 May 2020).

Samalis-Aldrich K and VonSpakovsky HA (2020) *Database Sells to 1,285 Proven Cases of Voter Fraud in America*. The Heritage Foundation. Available at: https://www.heritage.org/election-integrity/commentary/database-swells-1285-proven-cases-voter-fraud-america (accessed 17 May 2020).

Scott T (2019) *Why Electronic Voting Is Still a Bad Idea*. Available at: https://www.markpack.org.uk/160622/why-electronic-voting-is-still-a-bad-idea-tom-scott/ (accessed 19 May 2020).

Shamas M (2018) *FIDO Alliance Launches Biometrics Certification Program*. FIDO Alliance. Available at: https://fidoalliance.org/fido-alliance-launches-biometrics-certification-program/ (accessed 23 May 2020).

Simons BJ and Jones DW (2012) Internet voting in the U.S. *Communications of the ACM* 55: 68–77.

Smart-Card-Alliance (2008) *What Makes a Smart Card Secure*. Smart Card Alliance. Available at: https://www.securetechalliance.org/resources/lib/Smart_Card_Security_WP_20081013.pdf (accessed 19 June 2020).

Smith K (2018) *Pros and Cons of Internet Voting*. Goderich Signal Star. Available at: https://www.goderichsignalstar.com/2018/04/10/pros-and-cons-of-internet-voting/wcm/f31e6a24-6654-461f-30db-82e283adea10 (accessed 21 May 2020).

Stallings W (2014) *Network Security Essentials: Applications and Standards*. Upper Saddle River, NJ: Pearson Education, Inc.

Stilgherrian (2019) *Flaws Found in NSW Ivote System Yet Again*. IBM Security. Available at: https://www.zdnet.com/article/flaws-found-in-nsw-ivote-system-yet-again/ (accessed 30 October 2020).

Teague V (2019) *Faking an Ivote Decryption Proof*. Melbourne, Australia: University of Melbourne. Available at: https://people.eng.unimelb.edu.au/vjteague/iVoteDecryptionProof-Cheat.pdf (accessed 30 October 2020).

United States Census Bureau (2017) *Voting and Registration in the Election of November 2016*. Washington, DC: United States Census Bureau.

USEAC (2020) *Voting Equipment Registered Manufacturers*. USEAC. Available at: https://www.eac.gov/voting-equipment/registered-manufacturers (accessed 24 May 2020).

Vassil K, Solvak M and Vinkel Pet al (2016) The diffusion of internet voting. Usage patterns of internet voting in Estonia between 2005 and 2015. *Government Information Quarterly* 33(3): 453–459.

Venkatesh V, Brown SA and Bala H (2013) Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. *Management Information Systems Quarterly* 37: 21–54.

Venkatesh V, Morris MG, Davis GB, et al. (2003) User acceptance of information technology: Toward a unified view. *MIS Quarterly* 27, 425–478.

Vicens AJ (2019) *Online Voting Is a Really, Really Bad Idea*. Mother Jones. Available at: https://www.motherjones.com/politics/2019/11/online-voting-problems/ (accessed 19 May 2020).

Vinkel P (2011) Internet voting in Estonia. In: Laud P (ed.) *Information Security Technology for Applications*. Estonia: Lecture Notes in Computer Science, pp. 4–12.

Walker A (2019) *What Is Threat Modeling (+ Top Threat Model Examples)*. Available at: https://learn.g2.com/threat-modeling (accessed 6 June 2020).

Wang B, Zheng Y, Lou W, et al. (2015) DDoS attack protection in the era of cloud computing and Software-Defined Networking. *Computer Networks* 81: 308–319.

Wolchok S, Wustrow E, Isabel D, et al. (2012) Attacking the Washington DC Internet voting system. In: Keromytis AD (ed.) *Financial Cryptography and Data Security 16th International Conference*. Kralendijk, Bonaire: Springer, pp. 114–128.

Author biography

Jim Helm is a professor of Practice in the Information Technology Program within the Ira A. Fulton Schools of Engineering at Arizona State University. He has worked in various areas of cybersecurity and wireless communications for over 20 years in both industry and education.